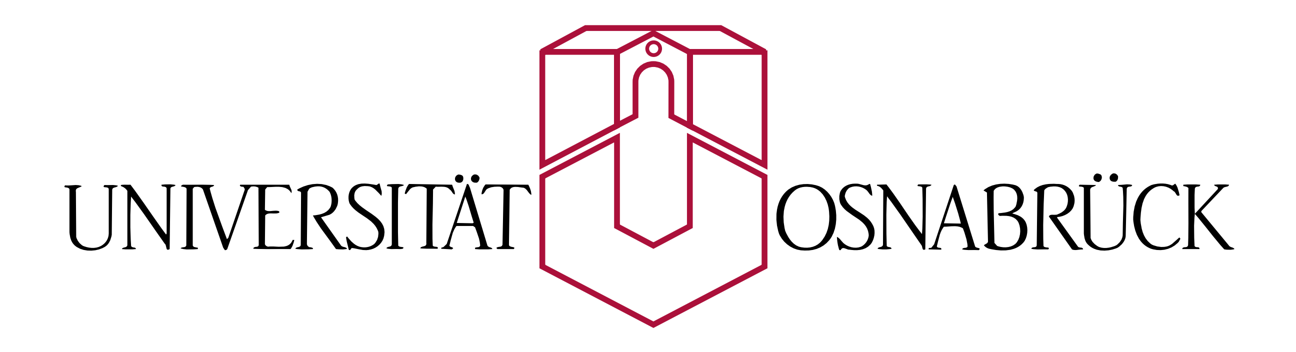



# Hypermediale Navigation in Vorlesungsaufzeichnungen

## Nutzung und automatische Produktion hypermedial navigierbarer Aufzeichnungen von Lehrveranstaltungen

**Dissertation**
**vorgelegt von**
**Robert Mertens**

**Fachbereich Mathematik/Informatik**
**Universität Osnabrück**

| | | |
|---|---|---|
| **Erstgutachter:** | **Prof. Dr. Oliver Vornberger** | **Universität Osnabrück** |
| **Zweitgutachterin:** | **Prof. Dr. Ute Schmid** | **Universität Bamberg** |

**2007**

**Abstract:** In the mid nineties, electronic lecture recording has emerged as a new area of research. The aim behind most early research activities in this field has been the cost-efficient production of e-learning content as a by-product of traditional lectures. These efforts have led to the development of systems that can produce recordings of a lecture in a fraction of the time and also for a fraction of the cost that other methods require for the production of similar e-learning content.

While the production of lecture recordings has been investigated thoroughly, the conditions under which the content produced can be used efficiently shifted into focus of research only recently. Employing lecture recordings in the right way is, however, crucial for the effectiveness with which they can be used. Therefore this thesis gives a detailed overview of archetypical application scenarios for recorded lectures. A closer examination of these scenarios reveals the importance of navigation in recorded lectures as a critical factor for teaching and learning success. In order to improve navigation in recorded lectures, a hypermedia navigation concept for recorded lectures is developed. Hypermedia navigation has proven a successful navigation paradigm in classic text- and picture-based media. In order to be adapted for time based media such as recorded lectures, a number of conceptual changes have to be applied. In this thesis, a hypermedia navigation concept is developed that tackles this problem by combining time- and structure-based navigation paradigms and by modifying existing hypermedia navigation facilities.

Even a highly developed navigation concept for recorded lectures can, however, not be put into practice efficiently when production costs of suitable recordings are too high. Therefore this thesis also shows that suitable lecture recordings can be produced with minimal production cost. This endeavour is realized by the implementation of a fully automatic production chain for recording and indexing lectures.

**Kurzzusammenfassung:** Seit Mitte der neunziger Jahre hat sich die Entwicklung von Vorlesungsaufzeichnungssystemen als Forschungsrichtung etabliert. Ziel der meisten Forschungsprojekte in diesem Bereich ist die kostengünstige Erzeugung von E-Learning Materialien als Nebenprodukt klassischer Vorlesungen. Dadurch sind im Vergleich zu herkömmlichen Ansätzen bei der Erzeugung von E-Learning Materialien deutliche Kostensenkungen erreicht worden.

In letzter Zeit rücken zunehmend die Rahmenbedingungen für den Einsatz von Vorlesungsaufzeichnungen in das Blickfeld der Forschung. Eine solche Perspektive erfordert eine ausführliche Betrachtung der Einsatzbedingungen, unter denen die erstellten Materialien genutzt werden. Im Rahmen der vorliegenden Arbeit werden daher verschiedene Nutzungsszenarien analysiert. Dabei zeigt sich, dass Möglichkeiten zur gezielten Navigation und Suche in Vorlesungsaufzeichnungen einen wesentlichen Erfolgsfaktor darstellen. Hauptziel der Arbeit ist deshalb die Entwicklung eines effektiven Navigationskonzeptes für Vorlesungsaufzeichnungen. Das Ergebnis der vorgestellten Forschungsarbeit lehnt sich an erprobte Navigationsansätze für konventionelle Hypermedien an. Ein besonderer Fokus liegt bei dieser Herangehensweise auf der Anpassung von Konzepten, die ursprünglich für text- und bildbasierte Medien konzipiert worden sind und nicht unverändert in zeitbasierten Medien eingesetzt werden können.

Die technische Umsetzbarkeit des erarbeiteten Konzeptes wird am Beispiel des Vorlesungsaufzeichnungssystems virtPresenter gezeigt. Das System vereint die automatisierte Erstellung von Vorlesungsaufzeichnungen mit der hypermedialen Aufbereitung und Darstellung der aufgezeichneten Inhalte. Auf diese Weise werden didaktische Vorteile einer hypermedial navigierbaren Aufzeichnung mit den Kostenvorteilen einer vollständig automatisierten Produktionskette verknüpft.

**Danksagung**

In der vorliegenden Arbeit werden die Ergebnisse mehrer Jahre Forschungstätigkeit am Zentrum virtUOS der Universität Osnabrück vorgestellt. Ohne die Unterstützung und Mithilfe vieler Beteiligter wäre die Fertigstellung dieser Arbeit und des zugrunde liegenden Projektes in der vorliegenden Form nicht möglich gewesen. Ich möchte mich daher an dieser Stelle bei allen Beteiligten bedanken.

An erster Stelle möchte ich mich bei Prof. Dr. Oliver Vornberger die Betreuung meiner Promotion und für die wertvolle Unterstützung bei einer Reihe von Projekten in diesem Zusammenhang danken.

Prof. Dr. Ute Schmid möchte ich für ihre Arbeit als Zweitbetreuerin und ihre Unterstützung bei der Vorbereitung meines Forschungsaufenthaltes an der University of Pittsburgh danken.

Dr. Andreas Knaden möchte ich für die Möglichkeit danken, mein Promotionsthema und meine Arbeit im Zentrum virtUOS miteinander zu verbinden. Darüber hinaus möchte ich mich bei ihm für viele wertvolle Hinweise und Anregungen im Zusammenhang mit der im Rahmen dieser Arbeit entstandenen vollständig automatisieren Produktionskette zur Erstellung von Vorlesungsaufzeichnungen bedanken.

Bei Prof. Dr. Peter Brusilovsky möchte ich mich für die Möglichkeit meines 2-monatigen Forschungsaufenthaltes am Personalized Adaptive Web Systems Lab an der University of Pittsburgh Anfang 2006 und die sehr gute Betreuung dieses Aufenthaltes danken. Den Mitarbeitern und Studierenden dort möchte ich für das gute Arbeitsklima danken, dem DAAD für die Finanzierung des Aufenthaltes.

Für die Mitarbeit an verschiedenen Softwarekomponenten des im Rahmen der vorliegenden Arbeit entwickelten virtPresenter möchte ich mich bei allen Beteiligten bedanken. Holger Schneider und Sergey Ishchenko haben bei der Entwicklung des Studierendeninterfaces großen Einsatz gezeigt. Torsten Klei hat die PowerPoint-Aufzeichnungskomponente und den Server zur Videokonvertierung implementiert. Philipp Hügelmeyer hat durch grundlegende Recherche zur Erzeugung von SVG aus PowerPoint und zur Steuerung der Aufzeichnungshardware zur Realisierung des Projektes beigetragen. Elmar Ludwig hat ein Kommunikationsprotokoll zur Ansteuerung der Aufzeichnungshardware entworfen und implementiert.

Auch für den Einsatz und die Erprobung des Systems möchte ich mich bei allen Beteiligten bedanken. An erster Stelle sind hier Nils Birnbaum und Rüdiger Rolf zu nennen. Nils Birnbaum hat in der Anfangsphase des Projektes unzählige Stunden mit dem Schneiden und Konvertieren von Videoaufzeichnungen zugebracht. Rüdiger Rolf hat seit 2006 den Einsatz des virtPresenter in einer Reihe von Lehrveranstaltungen koordiniert und begleitet. Er hat darüber hinaus eine Reihe wertvoller Hinweise zur Verbesserung verschiedener Softwarekomponenten geliefert. Auch bei den vielen Hilfskräften, die besonders in der Anfangsphase des Projektes viel technische Betreuungsarbeit geleistet haben, möchte ich mich hier bedanken. Den Dozenten und den Studierenden der aufgezeichneten Veranstaltungen möchte ich für Ihre Geduld bei der Arbeit mit vielen Prototypversionen des Systems danken.

Bei Clemens Gruber, Leonore Schulze und Carsten Wittenberg möchte ich mich für die Konzeption und Durchführung von entwicklungsbegleitenden Evaluationen und Nutzerbefragungen bedanken.

Bei Markus Ketterl möchte ich mich für die Erarbeitung und den Einsatz eines Konzeptes zur Erstellung von Podcasts aus Vorlesungsaufzeichnungen mit dem virtPresenter bedanken. Darüber hinaus möchte ich ihm und Rüdiger Rolf für die Bereitschaft danken, den virtPresenter nach Ende meiner Zeit beim Zentrum virtUOS weiterzuentwickeln und einzusetzen.

Ganz besonders herzlich möchte ich mich bei allen bedanken, die Teile dieser Arbeit in verschiedenen Stadien Korrektur gelesen haben. Namentlich sind dies Anja Krüger, Michael Elbers, Dr. Ralf Kunze, Dorothee Langfeld, Dr. Angela Brunstein und Dr. Martin Gieseking.
Meiner Frau danke ich für ihr Verständnis für die vielen Wochenenden, Urlaubstage und Abendstunden, die in diese Dissertation und die zugrunde liegende Implementation geflossen sind. Bei meiner Mutter bedanke ich mich für ihre Unterstützung.

# 1 Einleitung

Der Katalog der Lehrveranstaltungen, die an einer Universität oder Fachhochschule angeboten werden, wird im offiziellen Sprachgebrauch oft Vorlesungsverzeichnis genannt. Bereits diese Wortwahl impliziert die hervorstechende Rolle des Lehrveranstaltungstyps Vorlesung in der Hochschullehre. Auch die Tatsache, dass Vorlesungen fast ausschließlich von Professoren gehalten werden, unterstreicht die Bedeutung von Vorlesungen als Lehrform. Seminare, Übungen, Tutorien und Praktika werden im Gegensatz dazu oft von wissenschaftlichen Mitarbeitern und in Einzelfällen sogar von Studierenden geleitet.

Vorlesungen sind damit nicht nur sehr wichtig sondern in ihrer Durchführung auch vergleichsweise kostenaufwändig. Dieser Faktor wiederum führt dazu, dass Vorlesungen im Gegensatz zu anderen Lehrformen in der Regel nicht in mehreren Sitzungen parallel zueinander und meist auch nicht in jedem Semester angeboten werden. Aus der Kombination der niedrigen Angebotsfrequenz und der Relevanz von Vorlesungen zur Vermittlung von oft prüfungsrelevantem Lehrstoff ergeben sich zwei nennenswerte Konsequenzen. Zum einen ist besonders bei Vorlesungen, die von Studierenden mehrerer Fächer besucht werden, eine erhöhte Wahrscheinlichkeit von Terminkollisionen gegeben. Zum anderen sind die Hörerzahlen gerade in wichtigen Vorlesungen sehr hoch, wodurch die Möglichkeit zur Interaktion von Lehrenden und Studierenden faktisch nicht gegeben ist.

Eine Option, diesen Punkten entgegen zu wirken, stellt die Aufzeichnung von Vorlesungen auf elektronischen Medien und deren Einsatz in entsprechend ausgelegten didaktischen Szenarien dar. Zur Erstellung von elektronischen Vorlesungsaufzeichnungen existiert eine Reihe von Systemen. Je nach System können verschiedenste in der Vorlesung gezeigte Materialen wie beispielsweise Bildschirmpräsentationen mit Microsoft PowerPoint, Tafelanschrieb oder Overhead-Folien mit einer Audio- oder Videoaufzeichnung des Dozenten zu einem multimedialen Dokument verknüpft werden.

Durch die Einbindung der Video- oder Audioaufzeichnung wird das resultierende Dokument zu einem zeitbasierten Medium. Diese Eigenschaft bringt einerseits eine Reihe positiver Aspekte mit sich. So wirkt die Aufzeichnung lebendiger und authentischer als ein textbasiertes Skript. Auch wird alleine durch die Zeit, die der Dozent zur Abhandlung bestimmter Themen aufwendet oft deutlich, welche Passagen der Aufzeichnung wichtiger oder schwieriger sind als andere. Hinzu kommt die Tatsache, dass sich durch die Einbindung eines Audio- oder Videomitschnittes ein sehr hoher Automatisierungsgrad bei der Erstellung der durch die Aufzeichnung erzeugten Materialien und damit vergleichsweise niedrige Produktionskosten realisieren lassen. Die Nutzung derartiger Mitschnitte bringt jedoch nicht nur Vorteile mit sich. So sind Navigation und Suche in zeitbasierten Medien weitaus schwieriger als in anderen Medientypen. Wie in Kapitel 3 gezeigt wird, ist aber gerade die Möglichkeit zur effektiven Navigation und Suche für den sinnvollen Einsatz von Vorlesungsaufzeichnungen in vielen didaktischen Szenarien essentiell.

Ziel dieser Arbeit ist daher die Entwicklung eines effektiven Navigationskonzeptes für Vorlesungsaufzeichnungen. Dabei ist eine Reihe von Nebenbedingungen wie ökonomische Beschränkungen und die Möglichkeit zur Integration der entwickelten Lösung in den Hochschulalltag zu beachten. Im weiteren Verlauf der Einleitung werden zunächst in 1.1 einige Hintergrundinformationen zum Stand der Forschung auf dem Gebiet moderner Vorlesungsaufzeichnungssysteme gegeben. In 1.2 wird auf diesen Informationen aufbauend die konkrete Zielsetzung der Arbeit formuliert. 1.3 gibt einen kurzen Überblick über den Aufbau der Arbeit.

## 1.1 Ausgangslage und Motivation

Bereits seit den siebziger Jahren ist die Verwendung von Video als Medium zur Vermittlung von Lerninhalten ein Ziel im E-Learning [Chambers & Specher 1980]. Seit Mitte der neunziger Jahre existieren Forschungsprojekte, die sich mit der Entwicklung von Systemen zur automatischen Aufzeichnung von Vorträgen und Vorlesungen befassen [Hürst, Müller & Ottmann 2004]. Zu den ältesten Projekten zählen Authoring on the Fly (AOF) an der Albert-Ludwigs-Universität Freiburg und Classroom 2000 am Georgia Institute of Technology. In beiden Projekten wurde im Jahr 1995 mit der Arbeit an Vorlesungsaufzeichnungssystemen begonnen [Belenkaia, Lienhard, Mohamed & Ottmann 2004]. Im Laufe der Jahre sind viele weitere Projekte in verschiedenen Größenordnungen hinzu gekommen. Hierbei handelt es sich in einigen Fällen um allein stehende Diplomarbeiten und in anderen um groß angelegte Forschungsprojekte, in denen eine Reihe von Dissertationen entstanden sind.

In diesen Forschungsprojekten sind nicht nur Systeme entwickelt und eingesetzt worden. Es sind auch Erkenntnisse darüber gewonnen worden, welche Punkte bei der Entwicklung von Vorlesungsaufzeichnungen beachtet werden müssen und wie sich verschiedene Systeme in den Hochschulalltag integrieren lassen. Nachfolgend werden diese Erkenntnisse in sehr kurzer Form zusammengefasst, um eine Grundlage für die Formulierung der konkreten Zielsetzung der vorliegenden Arbeit in 1.2 zu erarbeiten.

### 1.1.1 Kostengünstige Erstellung von E-Learning Inhalten

Einer der wesentlichen Vorzüge von Vorlesungsaufzeichnungen gegenüber anderen Formen von E-Learning Materialien ist die vergleichsweise kostengünstige Produktion von Vorlesungsaufzeichnungen [Bacher 1999; Späth 2006; Chen, Ma, Meinel & Schillings 2003].

Kostenschätzungen zur Erstellung herkömmlicher E-Learning Inhalte wie computer- oder webbasierten Lehreinheiten variieren. In [Müller, Ottmann & Zhang 2002] werden die Kosten zur Erstellung eines webbasierten Kurses im Umfang von zwei Semesterwochenstunden auf 100 000 Euro geschätzt. Dies entspricht einem Entwicklungsaufwand von 100 Stunden für eine Stunde Inhalt. Eine Schätzung nach [Chen, Ma, Meinel & Schillings 2003] veranschlagt bis zu einem Mannjahr für die Erstellung eines webbasierten Kurses. Dies entspricht ca. 60 Stunden Arbeitsaufwand zur Erstellung von Inhalten für eine Stunde Lernzeit.

Bei der Erstellung von Vorlesungsaufzeichnungen kann dieser Arbeitsaufwand erheblich minimiert werden, da die Erstellung der Inhalte als Nebenprodukt einer regulär gehaltenen Lehrveranstaltung von statten geht [Bacher 1999]. Auch bei dieser Produktionsform ist jedoch zu differenzieren. So kann die manuelle Synchronisation von Audio- oder Videodateien mit in der Lehrveranstaltung gezeigten Folien 10 Stunden Arbeitszeit für eine einstündige Aufzeichnung betragen [Gütter, Neumann, Schill & Schreiter 1998]. Für das Projekt Knowledgebay wird die Bearbeitungszeit für die teilweise Synchronisation einer Aufzeichnung nur mit vortragsrelevanten Folien mit bis zu vier Stunden für eine Stunde aufgezeichnetes Material beziffert [Sporer, Köstlbacher & Erbacher 2005]. In Systemen wie LoD (Lectures on Demand) werden automatisch Ereignisse wie Folienwechsel protokolliert [Schertler & Bodendorf 2003]. Zur Erstellung eines synchronisierten Multimediadokumentes muss in derartigen Systemen nur der Anfang der Protokolldatei mit dem Anfang der Videoaufzeichnung abgeglichen werden. Bei einer Videoaufzeichnung, in der der Start der Präsentation zu erkennen ist, ist dieser Arbeitsschritt in weniger als einer halben Stunde möglich.[1] Sehr weit entwickelte Systeme wie AOF oder das TeleTeachingTool (TTT)

[1] In einigen Videoformaten wie beispielsweise dem oft zur Wiedergabe von Vorlesungsaufzeichnungen verwendeten Real Video Format ist die Kürzung eines Videos sehr rechenaufwändig und dauert daher einige Minuten. Eine kürzere Schätzung ist aus diesem Grund nicht realistisch.

synchronisieren bereits die Aufzeichnung der verwendeten Lehrmedien und des Videomitschnittes. Durch diese Vorgehensweise sind zur synchronisierten Wiedergabe keine manuellen Arbeitsschritte mehr nötig [Bacher 1999; Ziewer & Seidl 2002]. Abb. 1-1 gibt einen Überblick über die betrachteten Aufwandsabschätzungen. Zur Aufzeichnung von Videos wird jeweils ein zusätzlicher Arbeitsaufwand zum Auf- und Abbau der verwendeten Technik in Höhe von zwei Stunden berücksichtigt. Eine Abschätzung über weitere Arbeitschritte wie beispielsweise die Verteilung der erstellten Materialien an die Studierenden wird nicht berücksichtigt, da diese Schritte für unterschiedliche Medien in gleicher Weise erfolgen und darüber hinaus nicht Teil des Erstellungsprozesses sind.

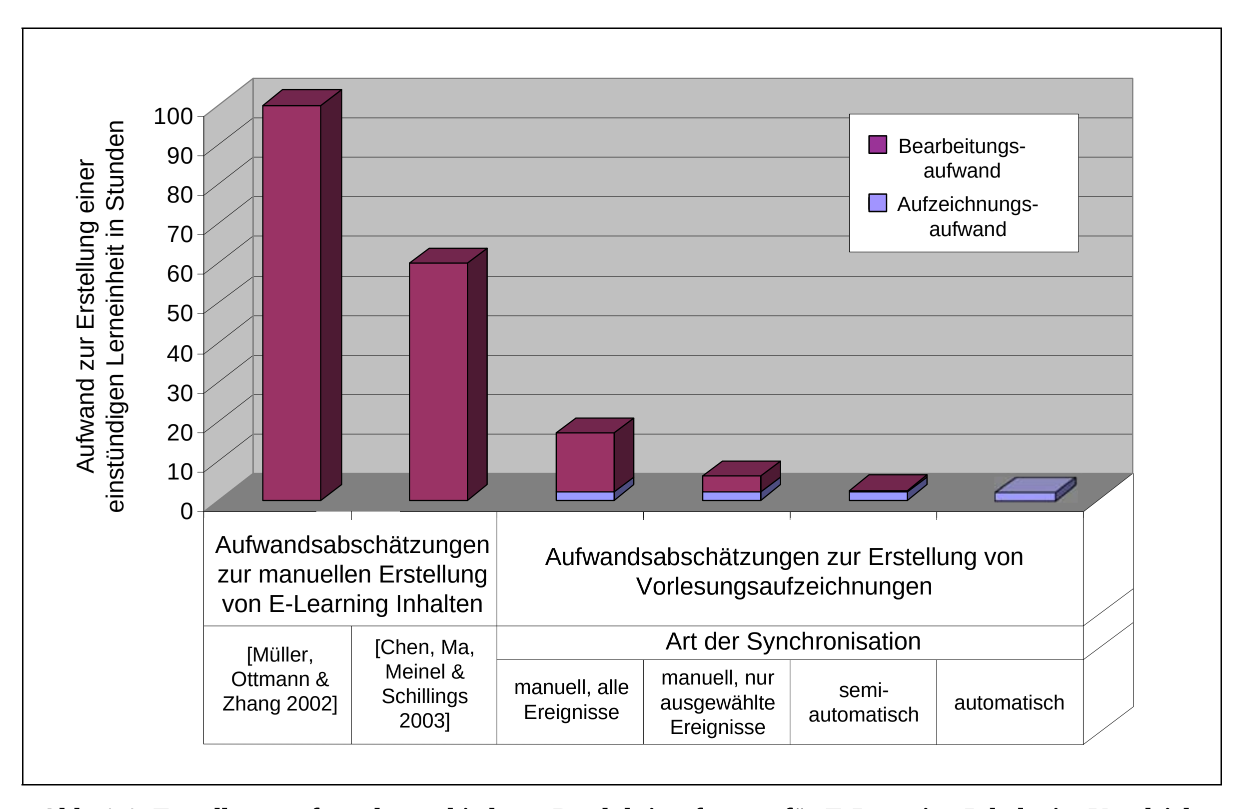



**Abb. 1-1: Erstellungsaufwand verschiedener Produktionsformen für E-Learning Inhalte im Vergleich**

Die Betrachtung der Aufwandsabschätzungen für die Produktion von Inhalten bei unterschiedlichen Produktionsformen ist vor allem unter dem Aspekt der Integration von E-Learning in den regulären Lehrbetrieb einer Hochschule wichtig. Hier ist das Kosten-Nutzen-Verhältnis bei der Erstellung von E-Learning Inhalten ein entscheidender Faktor [Müller, Ottmann & Zhang 2002]. Verschärft wird diese Problematik in vielen Fällen noch dadurch, dass E-Learning Materialien thematisch veralten und eine Aktualisierung der Inhalte erneute Produktionskosten mit sich bringt [Chen, Ma, Meinel & Schillings 2003].
Geringe Produktionskosten sind daher ein wichtiger Grund für den Einsatz und die Entwicklung von Vorlesungsaufzeichnungssystemen und dürfen bei der Entwicklung neuer Systeme nicht außer Acht gelassen werden.

### 1.1.2 Navigation in Vorlesungsaufzeichnungen

Klassische Formen der Navigation in zeitbasierten Medien wie das Vor- und Zurückspulen sind in Nutzungsszenarien wie der Arbeit mit Aufzeichnungen von Vorlesungen nur bedingt nützlich. Diese Navigationsformen haben sich zwar im Umgang mit Unterhaltungsmedien bewährt, sind jedoch im Kontext von Vorlesungsaufzeichnungen nicht hinreichend geeignet, da hier die Möglichkeit zur schnellen und effektiven Suche einzelner Abschnitte der Aufzeichnung in hohem Maße wichtig ist [Girgensohn, Boreczky & Wilcox 2001].

Besondere Bedeutung kommt dabei der Auffindbarkeit von semantisch relevanten Ereignissen wie Folien- oder Kapitelwechseln zu [Lauer & Trahasch 2005].
Gerade in neuartigen didaktischen Szenarien, in denen Vorlesungsaufzeichnungen als primäres Medium zur Wissensvermittlung eingesetzt werden, sind Möglichkeiten zur effektiven Navigation essentiell [Kumela, Watts & Adrion 2004]. Auch beim Einsatz von Vorlesungsaufzeichnungen als ergänzendes Lehrmaterial in traditionellen Vorlesungen spielen Navigationsmöglichkeiten in der Aufzeichnung eine tragende Rolle. In diesem Szenario werden Aufzeichnungen in erster Linie zur Lösung von Übungsaufgaben und für die Vorbereitung von Prüfungen und Klausuren verwendet [Zupancic & Horz 2002]. Die Studierenden sind daher im Groben mit dem Stoff der Vorlesung vertraut und benötigen Möglichkeiten, gezielt ausgewählte Passagen der Aufzeichnung anzusteuern [Hürst & Götz 2004].
Aus diesem Grund sind in den vergangenen Jahren eine Reihe von Ansätzen zur Navigation in Vorlesungsaufzeichnungen erarbeitet worden. Neben Ansätzen zur Navigation in Vorlesungsaufzeichnungen selbst sind auch Ansätze zur Suche und Navigation auf höheren Betrachtungsebenen entwickelt worden. Als Objekte in den Betrachtungsebenen können Sammlungen von multimedialen Dokumenten (beispielsweise alle Aufzeichnungen eines Kurses), einzelne Dokumente (Aufzeichnung einer Sitzung) und Abschnitte innerhalb einzelner Dokumente (Filmsequenzen) verstanden werden. Der Schwerpunkt der vorliegenden Arbeit liegt auf der letztgenannten Ebene. Die beiden anderen Ebenen werden im Folgenden kurz der Vollständigkeit halber diskutiert.
Zur Suche über Sammlungen von multimedialen Dokumenten ist eine abstrakte Beschreibung der Sammlung beispielsweise in Form von Metadaten nötig. An dieser Stelle können herkömmliche Navigations- und Suchverfahren eingesetzt werden.
Bei der Suche über einzelne Dokumente ist eine inhaltliche Analyse der Dokumente nötig. Zu diesem Zweck sind in einer Reihe von Forschungsprojekten spezielle Suchmaschinen entwickelt worden, die in der Vorlesung verwendete Folientexte analysieren. Beispiele hierfür sind in [Ziewer 2004], [Hürst 2002], [Denoue, Hilbert, Adcock, Billsus & Cooper 2005] und [Zhang, Rui, Crawford, & He 2005] zu finden. Im Rahmen von AOF ist sogar eine Suchmaschine entstanden, in der die Sprachspur der Aufzeichnung genutzt wird [Hürst 2004]. Auch wenn die Indizierung der Daten auf dieser Ebene teilweise auf anderen Techniken basiert als in herkömmlichen Suchmaschinen [Hürst 2002], weichen die verwendeten Benutzerschnittstellen kaum von denen herkömmlicher Suchmaschinen ab. Einzelne Systeme bieten die Möglichkeit, die Wiedergabe der Aufzeichnung gezielt zu einem zum Suchergebnis gehörigen Zeitpunkt in der Aufzeichnung zu starten [Hürst 2004] oder erlauben das Abspielen einer Audiodatei beim Überfahren von Links zu Suchergebnissen mit dem Mauszeiger [Denoue, Hilbert, Adcock, Billsus & Cooper 2005]. Der generelle Aufbau des Nutzerinterfaces bleibt jedoch größtenteils unverändert, da Suchergebnisse als statische Momentaufnahme gezeigt werden können. Probleme, die bei der zeitbasierten Navigation auftreten können, werden in das Interface zur Wiedergabe der Aufzeichnung verlagert.
Auf der letzten Betrachtungsebene ist nicht nur Indizierung und Suche ein Forschungsgegenstand. Auf dieser Ebene kommt auch der Zeit-Struktur-Dualismus von Vorlesungsaufzeichnungen zum Tragen. Herkömmliche Navigationsansätze sind hier daher nur unzureichend geeignet. Gleichzeitig ist die Suche und Navigation in den Aufzeichnungen für die meisten Einsatzszenarien von größerer Bedeutung als Suche und Navigation auf anderen Betrachtungsebenen, da hier inhaltlich gearbeitet wird. Bisherige Navigationsansätze sind meist entweder zeit- oder strukturbasiert. Zeitbasierte Ansätze beschränken sich in der Regel auf die Nutzung eines Sliderinterfaces zur Navigation in der Aufzeichnung. Hinzu kommen vereinzelt Systeme, in denen ein beschleunigtes Abspielen der Aufzeichnung möglich ist wie beispielsweise in [Li, Gupta, Sanocki, He & Rui 2000] beschrieben. Mit *visible scrolling*

wird das Sliderinterface durch eine instantane Vorschaufunktion erweitert, die zu jeder Sliderposition bereits bei der Bewegung des Sliders die zugehörige Folie anzeigt [Lauer & Ottmann 2002]. Ein ausführlicher Überblick über eine Reihe von Variationen des *visible scrolling* Ansatzes und über eine Reihe von Navigationsansätzen, die eher der Ebene Suche über einzelne Dokumente zuzuordnen sind, wird in [Hürst 2005] gegeben. Im Vergleich zu zeitbasierten Ansätzen existieren deutlich mehr unterschiedliche strukturbasierte Ansätze. Dort wird in der Regel die Folienstruktur des aufgezeichneten Vortrags genutzt. Beispiele sind Volltextsuche über Folientexte, eine zur Navigation nutzbare Folienübersicht und Möglichkeiten, einzelne Folien oder Animationsschritte zu überspringen. Auch diese Ansätze stellen wirkungsvolle Navigationshilfen dar, sind jedoch, wie im weiteren Verlauf dieser Arbeit gezeigt, in vielen Systemen nur unsystematisch zusammengewürfelt.
Eine Verbindung zwischen zeit- und strukturbasierten Ansätzen ist in bisherigen Systemen nur in sehr geringem Maße vorhanden. Durch eine kombinierte Nutzung von Zeit und Struktur bieten sich jedoch Potentiale zur Navigation in Vorlesungsaufzeichnungen, die durch rein zeit- oder rein strukturbasierte Ansätze nicht ausgeschöpft werden können [Mertens, Brusilovsky, Ishchenko & Vornberger 2006].

## 1.2 Zielsetzung der Arbeit

Hauptziel der vorliegenden Arbeit ist die Entwicklung eines effektiven Konzeptes zur Navigation in Vorlesungsaufzeichnungen. Die Realisierbarkeit des Konzeptes soll in einer Softwareanwendung demonstriert werden.
Darüber hinaus soll gezeigt werden, dass die Erstellung und der Einsatz von Vorlesungsaufzeichnungen, die mit der Softwareanwendung wiedergegeben werden können, sinnvoll in den Hochschulalltag integriert werden kann. Aufgrund der in 1.1.1 dargestellten Rahmenbedingungen muss die Aufzeichnung und Nachbearbeitung daher vollständig automatisch realisiert werden. Da die Integration von Vorlesungsaufzeichnungen in den Hochschulalltag kein rein technisches Problem darstellt, muss auch gezeigt werden, wie die Erstellung und der Einsatz von Vorlesungsaufzeichnungen organisatorisch und didaktisch entsprechend zu gestalten sind.

## 1.3 Aufbau der Arbeit

Neben Einleitung, Fazit und Ausblick lässt sich die Arbeit in drei wesentliche Schwerpunktbereiche unterteilen. Im Folgenden wird eine kurze Übersicht über diese Schwerpunktbereiche einschließlich der Zuordnung der Kapitel der Arbeit zu den Bereichen gegeben.

### 1.3.1 Grundlagen und Rahmenbedingungen

In den beiden folgenden Kapiteln der vorliegenden Arbeit werden einleitend Grundlagen und Rahmenbedingungen dargestellt, die bei der Entwicklung von Systemen zur Aufzeichnung und Wiedergabe von Vorlesungsmitschnitten zu beachten sind.
In **Kapitel 2** werden grundlegende Fragestellungen zur Produktion und Gestaltung von Vorlesungsaufzeichnungen behandelt. **Kapitel 3** ist der Darstellung verschiedener Einsatzszenarien von Vorlesungsaufzeichnungen gewidmet. Besonderes Augenmerk liegt dabei auf der Frage, welche Anforderungen sich aus den dargestellten Einsatzszenarien für die Navigation in Vorlesungsaufzeichnungen ergeben.

### 1.3.2 Entwicklung des Navigationskonzeptes

Im zweiten Schwerpunktbereich wird die Entwicklung des hypermedialen Navigationskonzeptes dargestellt. **Kapitel 4** dient der Analyse verschiedener Navigationsansätze hinsichtlich ihrer Eignung zur Verwendung in Vorlesungsaufzeichnungen und der

theoretischen Erarbeitung des Navigationskonzeptes. In **Kapitel 5** wird ein Überblick über Vorlesungsaufzeichnungssysteme mit für diese Arbeit relevanten Navigationsansätzen gegeben. Neben den Nutzerschnittstellen wird dabei auch der Produktionsprozess zur Erstellung von Vorlesungsaufzeichnungen mit den jeweiligen Systemen betrachtet. Die Umsetzung des erarbeiteten Konzeptes wird in **Kapitel 6** dargestellt.

### 1.3.3 Darstellung der Implementation

Die Implementation des im Rahmen der vorliegenden Arbeit entwickelten Systems zur Aufzeichnung und Wiedergabe von Vorlesungsmitschnitten wird in den Kapiteln 7 und 8 beschrieben. **Kapitel 7** dient dabei der Darstellung des Gesamtsystems einschließlich des vollständig automatisierten Produktionsprozesses. **Kapitel 8** beschreibt Details der Konvertierung der verwendeten Vortragsfolien in das webfähige Vektorgrafikformat SVG sowie die zur Nachbearbeitung und Aufbereitung der Folien zu Navigationszwecken nötigen Schritte.

### 1.3.4 Fazit und Ausblick

In **Kapitel 9** werden die im Rahmen der vorliegenden Arbeit gewonnenen Ergebnisse zusammengefasst. Abschließend wird ein Ausblick auf weitere Forschungsarbeiten gegeben, die durch die erarbeiteten Ergebnisse möglich geworden sind.

# 2 Grundlagen

Der Begriff Vorlesungsaufzeichnung umspannt ein weites Feld unterschiedlichster Endprodukte, die von reinen Audioaufzeichnungen bis hin zu multimedial aufbereiteten Dokumenten mit eingebetteten Videodateien reichen. Auch das Ausgangsmaterial einer Aufzeichnung wie etwa die Art der verwendeten Vortragsmaterialien und die Form der Vorlesung können variieren. Bei der Implementation eines Systems zur Aufzeichnung müssen all diese Faktoren berücksichtigt werden. Abhängig von der verfolgten Zielsetzung sind daher unterschiedliche Design-Entscheidungen zu treffen.

Wichtige Design-Entscheidungen, die im Rahmen dieser Arbeit getroffen werden müssen, umfassen die unterstützen Präsentationsformate, die aufzuzeichnenden Vortragsarten und die Frage, ob die Aufzeichnung ein Video beinhalten sollte. In diesem Abschnitt wird daher ein Überblick über Präsentationsformate in Vorlesungen und den aktuellen Stand der Forschung bezüglich des Nutzens der Verwendung von Videomitschnitten in Vorlesungsaufzeichnungen gegeben. Darüber hinaus werden aus pragmatischen Gründen heraus Ansätze zur automatisierten Produktion von Vorlesungsaufzeichnungen betrachtet. Die Motivation zur Automatisierung der einzelnen Schritte bei der Produktion von Vorlesungsaufzeichnungen ergibt sich dabei aus der Häufigkeit wiederkehrender und monotoner Arbeiten bei der Produktion und Bereitstellung von Vorlesungsaufzeichnungen im Regelbetrieb und den dadurch verursachten Kosten.

## 2.1 Produktion von Vorlesungsaufzeichnungen

Obwohl Vorlesungsaufzeichnungen im Vergleich zu herkömmlichem eLearning-Content mit vergleichsweise geringem Aufwand erstellt werden können (vgl. 1.1.1), ist die Produktion von videobasierten Vorlesungsaufzeichnungen im Vergleich zu rein audiobasierten Aufzeichnungen in der Regel mit erheblichem Mehraufwand verbunden [Knight, Steinbach & White 2002; Wolf, Putz, Stewart, Steinmetz, Hemmje & Neuhold 2004, Zhang, Rui, Crawford, & He 2005]. Dabei handelt es sich größtenteils um Arbeitsabläufe wie den Auf- und Abbau von Aufzeichnungstechnik oder Verwaltungsschritte wie den Upload der Aufzeichnung auf geeignete Webserver. Diese Schritte werden zwar meist von zusätzlichem Personal durchgeführt, können aber durch geeignete Automatisierungsmaßnahmen vermieden werden [Zhang, Rui, Crawford, & He 2005].

Der vorliegende Abschnitt untersucht die einzelnen Schritte bei der Produktion von Vorlesungsaufzeichnungen und erarbeitet Potentiale zur Automatisierung. Um eine systematische Betrachtungsweise der einzelnen Prozessschritte zu gewährleisten, orientiert sich die Gliederung des vorliegenden Abschnittes an dem in [Hürst, Müller & Ottmann 2004] vorgestellten Vier-Phasen-Modell zur Produktion und zum Einsatz von Vorlesungsaufzeichnungen. In dem Modell werden die vier Phasen Vorbereitung, Vortrag und Aufzeichnung, Nachbereitung und Nutzung unterschieden. Abb. 2-1 gibt einen schematischen Überblick über die einzelnen Phasen.

Zur Betrachtung der jeweiligen Produktionsschritte werden in den folgenden Unterabschnitten Lösungsansätze aus existierenden Systemen zusammengetragen. Im weiteren Verlauf der vorliegenden Arbeit wird in Kapitel 1 eine Implementation vorgestellt, die einige dieser Ansätze aufgreift und teilweise weiterentwickelt. Da der Schwerpunkt der vorliegenden Arbeit auf der Erstellung und Nutzung von online verwendbaren Vorlesungsaufzeichnungen liegt, wird die Produktion von Offline-Materialien im weiteren Verlauf dieses Abschnittes nicht behandelt.

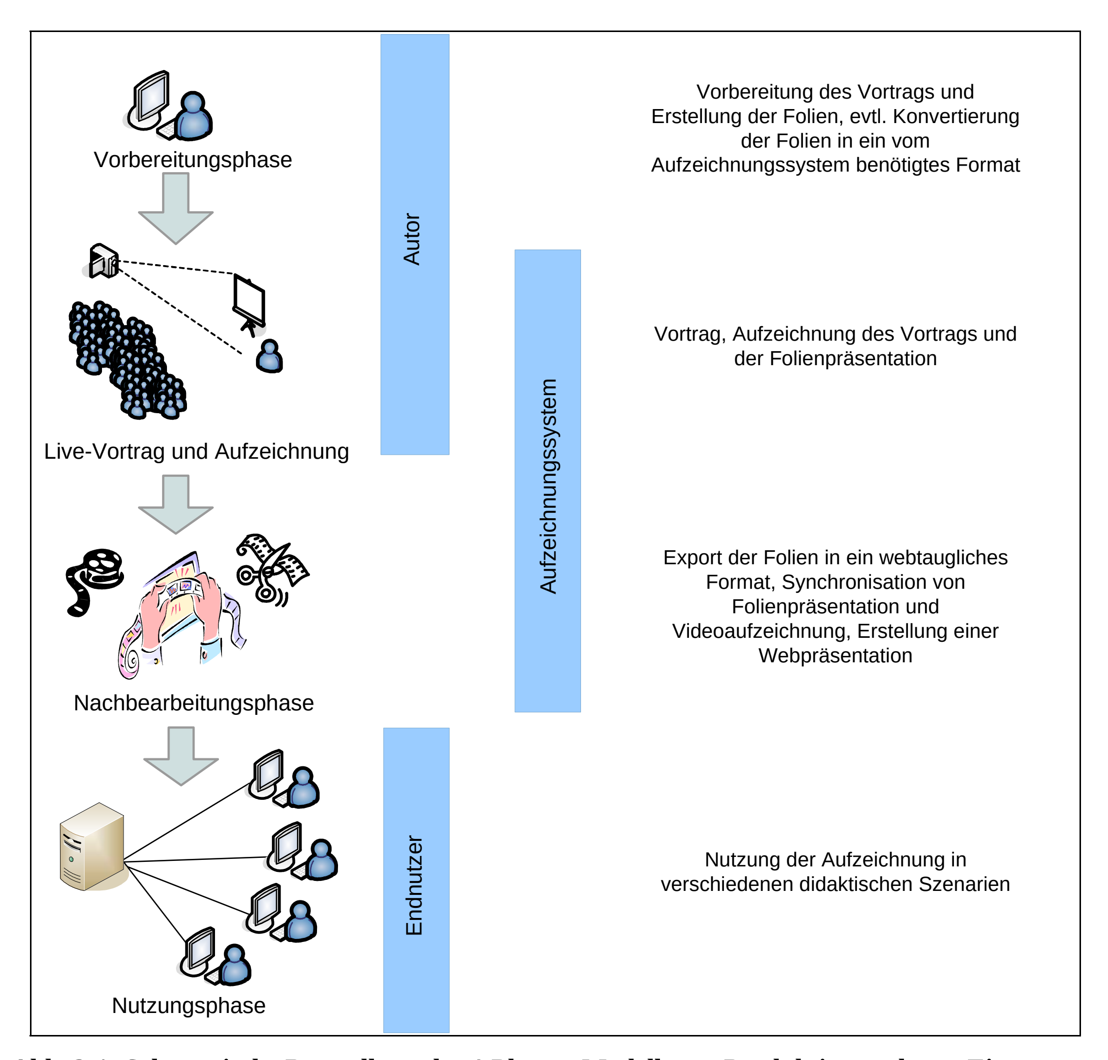



**Abb. 2-1: Schematische Darstellung des 4-Phasen-Modells zur Produktion und zum Einsatz von Vorlesungsaufzeichnungen nach [Hürst, Müller & Ottmann 2004]**

### 2.1.1 Vorbereitungsphase

Neben der Vorbereitung des Vortrags und eventuell verwendeter Vortragsmaterialien sind bei der Vorbereitung und Durchführung der Aufzeichnung einer Vorlesung weitere Schritte nötig. Diese Schritte variieren je nach Art der Aufzeichnung. So ist beispielsweise zur Aufzeichnung von Folienvorträgen in der Regel ein Import der Vortragsfolien in das jeweilige System durchzuführen [Brotherton 2001; Knipping 2005; Hürst, Müller & Ottmann 2004]. In den meisten Systemen gehen dabei jedoch Eigenschaften der Präsentation verloren [Lauer & Ottmann 2002]. Bei Systemen, die zusätzliches Personal zur Bedienung oder zum Auf- und Abbau von Aufzeichnungstechnik erfordern,[2] muss das Personal rechtzeitig angefordert und koordiniert werden. Wenn speziell ausgestattete Räume zur Aufzeichnung benötigt werden, ist im Vorfeld der Veranstaltung die Reservierung und Buchung dieser Räume zu tätigen.
Schritte wie der Import von Vortragsfolien können vielfach durch eine Anpassung des Aufzeichnungsprozesses vermieden werden (vgl. dazu 2.1.2.1). Organisations- und Verwaltungsschritte können beispielsweise durch die Anbindung des Aufzeichnungssystems an ein Lern-Management-System minimiert werden, wie in [Mertens, Knaden, Thelen & Vorn-

[2] So müssen beispielsweise in ePresence Folienwechsel unter gewissen Rahmenbedingungen manuell protokolliert werden (siehe 5.4). In anderen Fällen ist zusätzliches Personal zur Kameraführung beispielsweise zur Nachverfolgung von Tafelanschrieb erforderlich.

berger 2005b] dargestellt. In der Praxis kann eine derartige Kopplung jedoch leicht an einer Reihe administrativer Hindernisse scheitern.

### 2.1.2 Live Vortrag und Aufzeichnung

Zur Aufzeichnung von Vorlesungen existiert eine Reihe von verschiedenen Herangehensweisen, die in 2.2 in allgemeiner Form im Zusammenhang mit verschiedenen Präsentationsformaten diskutiert werden. Da die dort vorgestellten Ansätze zur Aufzeichnung bestimmter Präsentationsformate auch unterschiedliche Möglichkeiten zur technischen Umsetzung bieten, wird in 2.1.2.1 ein Überblick über Automatisierungspotentiale der jeweiligen Herangehensweisen hinsichtlich der Aufzeichnung gegeben. In 2.1.2.2 werden Ansätze zur Verringerung der zum Auf- und Abbau von Aufzeichnungstechnik benötigten Zeiten diskutiert.

#### 2.1.2.1 Aufzeichnung der Präsentationsmaterialen

Ein weit verbreiteter Ansatz ist die Aufzeichnung von Folienvorträgen mittels spezieller **Präsentationssoftware**. Beispiele hierfür sind die Systeme MANIC, AOF und eClass, die in den Abschnitten 5.1, 5.2 und 5.5 beschrieben werden. In diesen Systemen müssen die zum Vortrag verwendeten Medien entweder im System erstellt oder importiert werden. Wie im vorangehenden Abschnitt erwähnt, treten bei derartigen Ansätzen oft Einbußen in der Wiedergabequalität der verwendeten Präsentationsmedien auf (vgl. auch 5.2). Ein Weg, diesen Nachteil zu vermeiden, wird im Rahmen des Projektes Digital Lecture Halls (DLH) aufgezeigt. Dort wird die COM-Schnittstelle von Powerpoint (und anderen Windows-Programmen) genutzt, um Zeitpunkte von Folienwechseln festzustellen sowie Folientexte und Titel zu extrahieren [Mühlhäuser & Trompler 2002].
Bei der Aufzeichnung von Bildschirminhalten mit **Screengrabbing-Verfahren** kann beliebige Software zur Darstellung der zu vermittelnden Inhalte während der Vorlesung eingesetzt werden.[3] Dabei werden in unterschiedlichen Systemen sowohl hardwarebasierte wie bei LectureLounge [Wolf & Steinmetz 2002] als auch softwarebasierte Lösungen wie beim TeleTeachingTool [Ziewer 2004] eingesetzt. Ein Import der verwendeten Präsentationsmaterialien in ein proprietäres Datenformat ist in beiden Fällen nicht nötig. Die verwendete Bildschirmaufzeichnung ist allerdings mit dem Nachteil fehlender symbolischer Repräsentation [Lauer & Ottmann 2002] behaftet. So geht beispielsweise Information über den Text der verwendeten Folien verloren. Nachträglich arbeitende Bildanalysetechniken, wie im Rahmen des TTT eingesetzt [Ziewer 2004], ermöglichen es jedoch, diesen Nachteil zu einem großen Teil, wenn auch nicht perfekt, auszugleichen.
In einer weiteren Klasse von Ansätzen wird die Aufzeichnung nicht elektronisch gestützter Präsentationsformen wie **Tafelvorträgen** (vgl. 2.2.1) ermöglicht. Ein Beispiel für derartige Systeme ist das System E-Chalk. Dieses System nutzt zwar eine elektronische Tafel, erlaubt es Dozenten jedoch, den Arbeitsstil beizubehalten, den sie von normalen Tafelvorträgen gewohnt sind [Friedland 2006].
Darüber hinaus existieren **Allround-Lösungen**, in denen das verwendete Präsentationsmedium abgefilmt wird. Der Preis für die gewonnene Flexibilität ist hierbei in den meisten Fällen ein Qualitätsverlust in der Wiedergabe, der vor allem bei Verwendung elektronischer Präsentationsmedien deutlich wird. Hinzu kommt die Tatsache, dass eine nachträgliche Indizierung schwierig ist. [Mukhopadhyay & Smith 1999] stellen einen Ansatz vor, in dem abgefilmte Folienvorträge nachträglich mit den gezeigten Folien verglichen werden, um Folienwechsel und Folientitel erkennen und zuordnen zu können. Die Folien müssen dazu

[3] Derartige Verfahren können allerdings im Zusammenhang mit der Vorführung von Videodateien während der Vorlesung problematisch sein. In der Praxis hat sich gezeigt, dass sich die Qualität des Videos bei der Aufzeichnung deutlich verschlechtern kann.

nach dem Vortrag in das System importiert werden. Die Videoaufzeichnung wird mittels einer Projektionsmatrix transformiert, um die Verschiebung zwischen Aufnahmeperspektive und Draufsicht der Folien auszugleichen. Versuche mit Handschrifterkennung zur Indizierung von Tafelanschrieb bei abgefilmten Vorträgen sind beim augenblicklichen Stand der entsprechenden Technologien nicht ausreichend erfolgreich [Liu & Kender 2004]. Ein erhöhter Arbeitsaufwand ergibt sich in vielen Fällen durch die Notwendigkeit der Kameraführung beispielsweise zur Nachverfolgung relevanter Regionen einer Tafel. Diesem Problem kann jedoch durch automatisierte Kameraführung begegnet werden. Beispiele hierfür sind die Systeme *Virtual Videography* zur Aufzeichnung von Tafelvorträgen [Heck, Wallick & Gleicher 2007] und *AutoAuditorium* zur Aufzeichnung von Folienvorträgen [Bianchi 2004]. Darüber hinaus können Algorithmen zur automatischen Kameraführung auch in Verbindung mit anderen Aufzeichnungsformen wir folienbasierten Aufzeichnungssystemen eingesetzt werden, um den ästhetischen Eindruck der Aufzeichnung zu verbessern [Hartle, Bär, Trompler & Rößling 2005]. Ein Überblick über den aktuellen Stand der Forschung auf dem Gebiet automatischer Kameraführung zur Aufzeichnung von Vorlesungen ist in [Heck, Wallick & Gleicher 2007] zu finden.

#### 2.1.2.2 Auf- und Abbau von Aufzeichnungstechnik

Der Auf- und Abbau von Aufzeichnungstechnik sowie die damit verbundene Justierung von Audio- und Videotechnik stellen einen beträchtlichen Kostenfaktor bei der Produktion von Vorlesungsaufzeichnungen dar. Im Gegensatz zu Anschaffungskosten von Computer-Hardware und Aufzeichnungstechnik handelt es sich bei diesen Kosten um wiederkehrende Personalkosten, die bei jeder Aufzeichnung entstehen und sich daher nicht mit einer Erhöhung der Anzahl aufgezeichneter Veranstaltungssitzungen amortisieren [Zhang, Rui, Crawford, & He 2005]. Diese Tatsache bedeutet nicht, dass sich diese Kosten nicht auf anderem Wege, beispielsweise durch die Durchführung entsprechender Einsatzszenarien, amortisieren können (vgl. Kapitel 3). Insgesamt stellen derartige technische Betreuungskosten jedoch ein spürbares Hindernis bei der Produktion von Vorlesungsaufzeichnungen dar. So wird in einigen Systemen aufgrund des personellen Aufwands komplett auf die Aufzeichnung eines Videos verzichtet, wie beispielsweise in [Rohde & Thomas 2003] geschildert.

Aus diesem Grund sind verschiedene Ansätze zur Verringerung der laufenden Kosten entstanden, die sich in zwei Gruppen unterteilen lassen. Eine Gruppe von Systemen setzt hierbei auf die feste **Integration der Aufzeichnungstechnik in ausgewählte Hörsäle**. Beispiele hierfür sind in [Mühlhäuser & Trompler 2002] und [Cooperstock 2001] beschrieben. In vielen Fällen sind derartige Ansätze mit einer weiteren Automatisierung der Hörsäle verbunden. Auch die Nutzung herkömmlicher Multimediahörsäle, die beispielsweise zum Einsatz von Videokonferenzen in der Lehre bestimmt sind, ist mit einer Reihe von Vorteilen verbunden. In diesem Fall ist Ton- und Videotechnik bereits vorhanden und die Anpassung an die zur Vorlesungsaufzeichnung genutzten Systeme bedeutet oft nur geringe einmalige Mehrkosten.

Die zweite Gruppe von Systemen nutzt **mobile Aufzeichnungstechnik**, die in kurzer Zeit auf- und abgebaut werden kann. Beispiele sind LectureLounge [Wolf, Putz, Stewart, Steinmetz, Hemmje & Neuhold 2004] und die an der Leibniz Universität Hannover entwickelte UbiMotion [Breitner & Breuer 2006]. Die Vorteile mobiler Lösungen liegen in niedrigeren Sicherungskosten gegen Diebstahl und Vandalismus, deutlich geringeren Anschaffungskosten sowie einer größeren Flexibilität bei der Raumauswahl der aufzuzeichnenden Veranstaltungen. Nachteile sind neben einem höheren Verwaltungsaufwand zusätzlicher, wenn auch minimierter, Betreuungsaufwand und mögliche Einbußen bei der Aufzeichnungsqualität wie beispielsweise Ton- und Beleuchtungsprobleme entgegen.

### 2.1.3 Nachbearbeitungsphase

Nachdem die Aufzeichnung abgeschlossen ist, werden die aufgezeichneten Medienströme in ein zur Wiedergabe geeignetes Format überführt und in den meisten Fällen in eine zur Navigation geeignete Oberfläche eingearbeitet. Diese Vorgänge sind in vielen Systemen vollständig automatisiert [Lauer und Ottmann 2002].

Darüber hinaus sind in der Regel weitere, manuell durchzuführende Schritte nötig, wie das Hochladen der erstellten Aufzeichnung auf geeignete Server und der Eintrag zugehöriger Links beispielsweise in kursspezifische Webseiten. Auch hierzu existiert eine Reihe von Lösungsansätzen. So werden in einem in [Knight, Steinbach & White 2002] beschriebenen System Vorlesungsaufzeichnungen aus Hörsälen über Nacht automatisch auf einen Webserver transferiert. In VClass können Bildschirmaufzeichnungen per Drag and Drop auf eine kursspezifische Webseite geladen werden und werden automatisch in ein streambares Format konvertiert [Baez-Franceschi, Dinshaw, Evans & Van Nieuwenhuise 2006]. Auch in ePresence besteht die Möglichkeit, Vorlesungsaufzeichnungen automatisch auf eine Webseite zu laden (vgl. 5.4).

### 2.1.4 Nutzungsphase

In der Nutzungsphase werden die erstellten Inhalte von Studierenden in verschiedenen Lernszenarien genutzt (vgl. Kapitel 3). Bei der Verwendung von online Distributionswegen ist hier kein nennenswerter Arbeitsaufwand gegeben. Lediglich beim Einsatz der Aufzeichnungen in einer Lehrveranstaltung kann zusätzlicher Arbeitsaufwand entstehen. So kann es sinnvoll sein, nur bestimmte Ausschnitte einer Aufzeichnung zur Verfügung zu stellen, auf diese zu verweisen oder sie nachträglich mit Anmerkungen zu versehen. Hierzu existieren verschiedene Ansätze. Im Interactive Shared Educational Environment können Kommentare in einem integrierten Chat mit Verweisen auf beliebige Zeitpunkte in einer Videodatei versehen werden [Mu 2005]. In [Miyahara, Kaiya & Kaijiri 2002] wird ein System beschrieben, das es Studierenden erlaubt, Ausschnitte aus Videodateien in einer externen Umgebung zu sammeln und neu zu kombinieren. In einer Erweiterung von AOF (siehe 5.2) können Studierende in aufgezeichneten Vorlesungen einzelne Folien mit eigenen handschriftlichen Kommentaren versehen [Lienhard & Lauer 2002]. In einer anderen Erweiterung zu AOF sind sogar mündliche Annotationen durch die Studierenden möglich [Lauer & Busl 2007].

### 2.1.5 Fazit

Der vorliegende Abschnitt hat eine Reihe von Automatisierungspotentialen in den einzelnen Phasen aufgezeigt. Besonders in der Aufzeichnungsphase (2.1.2) und der Nachbearbeitungsphase (2.1.3) können durch eine entsprechende Implementation in erheblichem Maße personelle Ressourcen eingespart werden. Vor allem durch die feste Integration von Aufzeichnungstechnik in Hörsäle und durch eine Automatisierung des Hochladens der aufgezeichneten Dateien auf geeignete Server samt Verlinkung der Aufzeichnung auf kursspezifischen Webseiten können unnötige Arbeitsschritte eliminiert werden. Eine umfassende Darstellung der Umsetzung und Anpassung der in diesem Abschnitt dargestellten Automatisierungskonzepte ist in Kapitel 7 zu finden. Aufgrund der Tatsache, dass einzelne Automatisierungskonzepte nicht universell anwendbar sind oder der im Rahmen dieser Arbeit verfolgten Zielsetzung nicht entsprechen, wird dabei nur eine Auswahl der hier dargestellten Ansätze berücksichtigt.

## 2.2 Vortragsarten und Präsentationsformate

Obwohl Vorlesungen in einigen Fächern im wortwörtlichen Sinne nur aus dem Akt des Vorlesens eines Buches oder Vorlesungsskriptes bestehen, werden in der universitären Lehre in der Regel eine Reihe von vortragsunterstützenden Medien eingesetzt. Traditionell handelt es sich dabei um Tafel und Kreide, seit ein paar Jahrzehnten auch um Folie und Overheadprojektor. Auch heute sind Tafel und Overheadprojektor an der Universität Osnabrück [Ollermann, Gruber & Hamborg 2003] und sicherlich auch an den meisten anderen Universitäten die am weitesten verbreiteten Vortragsunterstützungsmittel. Mit dem Aufkommen von Präsentationssoftware gegen Ende der achtziger Jahre [Endicott 2000] hat allerdings auch der Einzug elektronischer Präsentationen in die Hochschullehre begonnen. Mittlerweile werden nicht nur elektronische Präsentationsmedien wie Powerpoint [Lauer & Ottmann 2002] sondern auch speziell entwickelte didaktisch motivierte Animationsprogramme wie beispielsweise in [Brusilovsky & Su 2002], [Ziewer 2001] oder [Rößling 2004] beschrieben als vortragsunterstützende Medien in Vorlesungen eingesetzt. Darüber hinaus können in Lehrveranstaltungen auch beliebige Softwareanwendungen, Diamaterial oder Lehrfilme vorgeführt werden. In naturwissenschaftlichen Fächern kommt die Vorführung von Versuchen hinzu. Abb. 2-2 gibt einen Überblick über für Vorlesungsaufzeichnungen relevante Vortragsarten nach [Liu & Kender 2004].

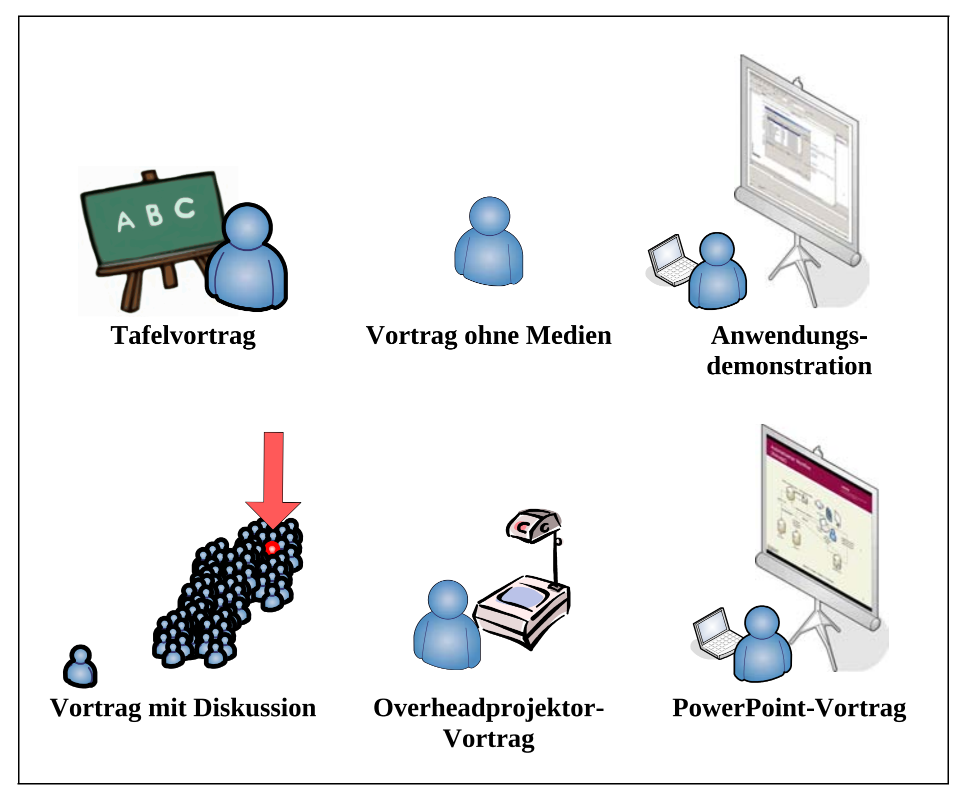



**Abb. 2-2: Im Kontext von Vorlesungsaufzeichnungen relevante Vortragsarten nach [Liu & Kender 2004].**

Bei der Aufzeichnung von Vorlesungen stellt diese Vielfalt von unterschiedlichen Medien zusammen mit der Möglichkeit von Rückfragen aus dem Publikum[4] eine Reihe von

[4] Nach [Bacher & Ottmann 1995] treten derartige Zwischenfragen gerade in größeren Lehrveranstaltungen selten bis gar nicht auf. In Anbetracht der Tatsache, dass Vorlesungsaufzeichnungen in dem meisten Fällen eher in größeren Veranstaltungen durchgeführt werden, ist der Punkt Zwischenfragen daher nicht unbedingt

Anforderungen an das jeweilige Aufzeichnungssystem [Liu & Kender 2004]. Einige Ansätze wie [Liu & Kender 2004] oder [Mukhopadhyay & Smith 1999] begegnen diesem Sachverhalt mit der Konzeption von Universalsystemen, die die Aufzeichnung von beliebigen Vorlesungstypen erlauben. Bei anderen Systemen wird aus den teilweise divergierenden Anforderungen [Lauer & Ottmann 2002] die Konsequenz gezogen, jeweils nur spezielle vortragsunterstützende Medien wie elektronische Tafeln in E-Kreide [Friedland 2006], HTML in MANIC (vgl. 5.1) oder PowerPoint in eTEACH (vgl. 5.6) zu verwenden. Bei einigen dieser Systeme können zusätzlich zur Präsentation der eingebundenen Medien auch handschriftliche Annotationen aufgezeichnet werden. Mit Bildschirmaufzeichnungssystemen wie dem TeleTeachingTool (vgl. 5.3), tele-TASK [Meinel & Schillings 2002], Lecture-Lounge [Wolf, Putz, Stewart, Steinmetz, Hemmje & Neuhold 2004] oder ProjectorBox [Denoue, Hilbert, Adcock, Billsus & Cooper 2005] können beliebige Bildschirminhalte synchron mit einer Videoaufzeichnung des Vortrags aufgezeichnet werden.
Bei der Wahl der Präsentationsformate, die durch das im Rahmen dieser Arbeit entwickelten Vorlesungsaufzeichnungssystem unterstützt werden, stand der Schwerpunkt der Arbeit auf Navigation in Vorlesungsaufzeichnungen im Vordergrund. Vor allem inhaltliche Navigation kann nur dann zuverlässig realisiert werden, wenn auch Zeitmarken zur inhaltlichen Struktur des Vortrags, wie beispielsweise an Folienwechselpunkten erkennbar, während der Aufzeichnung protokolliert werden [Lauer & Ottmann 2002].[5] Als Präsentationskomponente für das im Rahmen der Arbeit entwickelte Aufzeichnungssystem wurde aus diesem Grund Microsoft PowerPoint gewählt. Ein weiterer Grund ist, dass PowerPoint ein weit verbreitetes elektronisches Präsentationsformat ist, das neben Folienwechseln auch flexible Möglichkeiten zur Animation durch eine Vielzahl von Effekten bietet. Durch die Nutzung derartiger Animationseffekte konnte unter anderem eine sehr feingranulare strukturelle Navigation und die direkte zeitliche Ansteuerbarkeit von animierten Folienbestandteilen in der Aufzeichnung realisiert werden (vgl. 6.2.2).
Da der PowerPoint-Vortrag sich als Lehrform substantiell vom Tafelvortrag und ähnlichen Vortragsarten, in denen die Vortragsarbeit des Dozenten zu großen Teilen aus dem Niederschreiben von Informationen besteht, unterscheidet, werden im Folgenden beide Lehrformen und deren Charakteristika sowie mögliche Mischformen diskutiert. Dabei werden vor allem Einsatzrahmen und Einschränkungen der jeweiligen Präsentationsformen dargestellt, um neben einem Abriss der didaktischen Hintergründe auch eine Grundlage für die Darstellung und Diskussion verschiedener Einsatzszenarien von Vorlesungsaufzeichnungen in Kapitel 3 zu liefern.

### 2.2.1 Tafelvorträge und verwandte Präsentationsformen

Neben dem traditionellen Tafelvortrag gibt es eine Reihe von verwandten Präsentationsformen, die sich vom Tafelvortrag nur durch das verwendete Medium unterscheiden. So ist der didaktische Unterschied zwischen herkömmlicher Tafel, elektronischer Tafel und leeren Overheadfolien prinzipiell zu vernachlässigen. In all diesen Vortragsformen werden die zu vermittelnden Inhalte auf einem leeren Medium vollständig neu entwickelt.[6] Von [Knipping 2005] werden Flexibilität und die Möglichkeit zur Improvisation bei der Vortragsgestaltung, die Begrenzung der Vortragsgeschwindigkeit durch die handschriftliche Begleitung des

relevant. Erfahrungen an der Universität Osnabrück haben darüber hinaus gezeigt, dass gute Tonaufzeichnungen von Zwischenfragen in der Regel ausreichen und ein Kamerafokus nicht erforderlich ist.

[5] Das TeleTeachingTool kann beliebige Bildschirminhalte aufzeichnen. In diesem System ist eine heuristische Erkennung von Folienwechseln implementiert (vgl. 5.3). Kleinere Änderungen von Folien, wie beispielsweise durch Animationsschritte bedingt, sind mit derartigen Heuristiken jedoch nicht festzustellen.

[6] Mischformen, in denen vorbereitete Folien mit handschriftlichen Anmerkungen ergänzt werden, werden in 2.2.3 kurz andiskutiert.

Vortrags und das Hervorheben von wichtigen Punkten durch Einkreisen, Unterstreichen oder ähnliche Markierungen als Vorteile dieses Vorgehens angeführt. Ein weiterer Vorteil ist die einfache Verwendung beliebiger Schriftzeichen und Formeln, die in elektronischen Präsentationen oder gedrucktem Text oft nur umständlich durch Formeleditoren oder komplizierten Schriftsatz verwendet werden können.
Vor allem Flexibilität und Improvisation sind dabei Alleinstellungsmerkmale von Tafelvorträgen und ihnen verwandten Präsentationsformen einschließlich der in 2.2.3 dargestellten Mischformen. Eine Verlangsamung der Vortragsgeschwindigkeit kann bei Verwendung entsprechender Präsentationssoftware auch durch die Anpassung der Geschwindigkeit von Folienübergängen oder Animationen erreicht werden. Hinzu kommt, dass gerade in Formalwissenschaften wie beispielsweise der Mathematik komplexe Sachverhalte sehr kompakt dargestellt werden können, so dass die Vortragsgeschwindigkeit kaum durch den Akt des Anschreibens dieser Sachverhalte an eine Tafel gemindert wird. Die Markierung von relevanten Punkten ist beispielsweise durch neuere Kategorien von Animationseffekten in PowerPoint XP, die neben dem Erscheinen von beliebigen Folienobjekten wie Text- oder Grafikbausteinen auch deren Verschwinden und Verschieben zulassen, in elektronischen Präsentationen durchaus möglich. Wie im folgenden Abschnitt dargestellt, kann dabei sogar ein höherer Grad an didaktischer Qualität erzielt werden als mit konventionellen Mitteln. Ein Nachteil von Tafelvorträgen und verwandten Präsentationsarten ist die Abhängigkeit von der Handschrift des Dozenten. Bei schlechter Lesbarkeit der Handschrift des Dozenten ist der Tafelanschrieb nur von mäßigem Nutzen.

### 2.2.2 Elektronische Folienpräsentationen

Folienpräsentationen können in einer Reihe von elektronischen Präsentationsformaten wie Macromedia Flash, PowerPoint oder SVG erstellt werden [Casado & Torres 2004]. Alle der voran genannten Formate unterstützen die Animation (Bewegung entlang von vordefinierten Pfaden, Erscheinen, Verschwinden usw.) von einzelnen Folienbestandteilen und stellen Medien zur vorbereiteten und strukturierten Präsentation von Inhalten dar. Weitere verbreitete elektronische Präsentationsformate sind beispielsweise FoilTex oder Open Office Impress. PowerPoint ist sehr einfach zu bedienen ist und verfügt trotzdem über eine Vielzahl von Effekten zur didaktisch wirkungsvollen Gestaltung von Präsentationen verfügt. Darüber hinaus ist es die momentan am weitesten verbreitete Präsentationserstellungssoftware. Argumentationen gegen den Einsatz von PowerPoint in der Lehre wie [Knipping 2005] berufen sich oft auf die Kritik, die der emeritierte Yale-Professor Edward Tufte beispielsweise in [Tufte 2003] an PowerPoint als Präsentationsmedium übt. Tufte kritisiert vor allem die geringe Informationsdichte und die hierarchische Struktur von Stichwortlisten. Darüber hinaus richten sich die Ausführungen Tuftes gegen die in PowerPoint vorgegebene Form der Datenvisualisierung in Diagrammen. Explizit argumentiert Tufte auch gegen den Einsatz von PowerPoint an Grundschulen. Dem ist entgegenzuhalten, dass die Lehrsituation an Universitäten sehr häufig eine sehr hohe Informationsdichte beinhaltet und dass PowerPoint weitaus mehr als ein Werkzeug zur Darstellung statischer Textseiten und Diagramme ist. So lässt Tufte beispielsweise den didaktischen Wert von graphischen Darstellungen und Animationen vollkommen außer acht. Eine direkte Auseinandersetzung mit der Kritik von Tufte ist in [Simmons 2004] zu finden. Die Debatte um PowerPoint wird darüber hinaus von vielen Akteuren und teilweise auch recht emotional geführt. [Barrett 2004] gibt einen ausführlichen Überblick über die einzelnen Positionen und liefert darüber hinaus Hinweise zur sinnvollen Erstellung von PowerPoint-Präsentationen.
Unabhängig von dieser Debatte setzen viele Lehrende PowerPoint als vortragsunterstützendes Medium in Vorlesungen ein. [Eskicioglu & Kopec 2003] nennen den höheren Strukturierungsgrad von PowerPoint-Präsentationen und die bessere Lesbarkeit als Kriterien,

die für den Einsatz von PowerPoint in der Lehre sprechen. Durch die Verwendung von PowerPoint ist der Dozent gezwungen, den Ablauf des Vortrags im Vorfeld zu planen und durch rechnergestützte Erstellung der Präsentation entsteht (zumindest optisch) qualitativ hochwertiges Unterrichtsmaterial. Im Kontext von Vorlesungsaufzeichnungen hat eine vergleichende Untersuchung von [Brotherton & Abowd 2004] erbracht, dass Vorlesungsaufzeichnungen, in denen vorbereitete PowerPoint-Folien verwendet werden, von Studierenden bevorzugt genutzt werden.[7]

Neben Lesbarkeit und Strukturierung ist vor allem die Möglichkeit zum Einsatz von Animationen einer der entscheidenden Vorteile von PowerPoint. Durch Animationen kann beispielsweise der schrittweise Aufbau von Textbausteinen oder auch Grafiken realisiert werden, eine Technik, die aus lern- und aufmerksamkeitspsychologischer Sicht durchaus sinnvoll ist [Weidenmann 2002]. Darüber hinaus lassen sich einzelne Punkte in Grafiken oder Texten hervorheben und Abläufe können visualisiert werden. Auch die Wiederverwendbarkeit von komplexen Grafiken, in denen beispielsweise nur ein Detail verändert werden muss, ist gegeben. Dies stellt besonders dann einen Vorteil dar, wenn die entsprechende Grafik in einer anderen Sitzung der Veranstaltung verwendet worden ist. Animationen haben sich vor allem in Teilbereichen der Naturwissenschaften und der Informatik als effektives Mittel zur Wissensvermittlung gezeigt [Brusilovsky 2002]. [Huk, Steinke & Floto 2003] haben in einer Vergleichsuntersuchung gezeigt, dass Animationen im Zusammenhang mit visuellen Signalen, die relevante Teilaspekte hervorheben, signifikant bessere Lernleistungen ermöglichen.

Prinzipiell lassen sich mit neueren PowerPoint-Versionen fast alle Animationseffekte erzeugen, die mit Formaten wie Macromedia Flash oder SVG möglich sind. PowerPoint-Präsentationen sind jedoch erheblich einfacher zu erstellen und sind explizit zur Vortragsunterstützung konzipiert. Animationseffekte und Zurück-/Weiterblättern in der Präsentation kann daher ohne zusätzliche Programmierarbeit per Tastendruck gesteuert werden.[8]

Fast alle wichtigen Systeme zur Vorlesungsaufzeichnung unterstützen den Import von PowerPoint-Foliensätzen oder sind sogar explizit zur Aufzeichnung von PowerPoint-Vorträgen konzipiert (vgl. Kapitel 3). Meist werden Animationen dabei jedoch gar nicht oder nur fehlerhaft abgebildet. Teilweise ist dies sicherlich historisch bedingt, da Animationen erst 1995 in PowerPoint eingeführt wurden und viele Animationseffekte erst in weiteren Versionen folgten [Endicott 2000].

### 2.2.3 Mischformen

Viele Vorlesungsaufzeichnungssysteme, die zunächst auf der Metapher des Tafelvortrags oder verwandter Präsentationsformen basierten, wurden nachträglich um Importmöglichkeiten für PowerPoint-Foliensätze erweitert. Gründe hierfür sind beispielsweise die große Menge von bereits in PowerPoint vorhandenem Material [Knipping 2005] oder die Tatsache, dass viele Benutzer PowerPoint-Folien auf anderen Wegen importieren [Brotherton 2001]. Das Aufzeichnungssystem AOF (vgl. 5.2) wurde anfänglich zur Aufzeichnung von PostScript-Folien, handschriftlichen Annotationen und einer zugehörigen Audio- oder Videospur entwickelt [Bacher & Ottmann 1995]. Als Resultat dieser Mischung aus

[7] Interessanterweise war das in der Studie verwendete System auf handschriftliche Annotationen und nicht auf PowerPoint optimiert [Brotherton 2001].

[8] Eine derartige Kontrolllogik ist nicht zwingend nötig. So stellt beispielsweise [Voswinckel 2003] ein strukturorientiertes Präsentationsprogramm auf SVG-Basis mit einer Baumnavigation vor. Nichtsdestotrotz wird eine einfache Bedienbarkeit des Präsentationswerkzeuges von den meisten Nutzern bevorzugt [Hürst & Meyer 2004]. In SVG ist eine Steuerung über Tasten nicht integriert, lässt sicht jedoch mit vertretbarem Aufwand in selbst geschriebenen Präsentationswerkzeugen implementieren.

handschriftlichen Anmerkungen und vorbereiteten Folien ähneln viele der mit diesen Systemen erstellten Aufzeichnungen einer heute an Universitäten nur noch sehr vereinzelt zu findenden Präsentationsart, dem Overheadprojektor-Vortrag mit teilweise vorbereiteten Folien. Dozenten können in dieser Präsentationsform Folien während des Vortrags vervollständigen. Vorteilig ist eine solche Mischung von Folien und handschriftlichen Annotationen vor allem aufgrund der höheren Flexibilität, die dadurch entsteht, die Folien auch während des Vortrags noch ergänzen zu können [Lauer & Ottmann 2002]. Dozenten können dabei nicht nur Folien ergänzen, Unterstreichungen vornehmen und Zeichnungen während des Unterrichts entwerfen sondern auch mit anfänglich leeren Vorlagen improvisieren oder beispielsweise mathematische Beweise entwickeln [Mohamed, Belenkaia & Ottmann 2004].
Ein Nachteil derartiger Präsentationsformen ist, dass, auch aufgrund technischer Unzulänglichkeiten vieler Systeme, Annotationen oft anstelle von Animationen verwendet werden (vgl. 4.2.2.2). Der didaktische Vorteil von Animationen geht daher bei dieser Präsentationsform in den meisten Fällen verloren. Auch weitere Vorteile von elektronischen Folienpräsentationen wie die bessere Lesbarkeit des Materials und der hohe Vorstrukturierungsgrad des Vortrags können bei dieser Mischform wegfallen.

### 2.2.4 Diskussion und Vergleich der Präsentationsformate

Präsentationsformen wie der Tafelvortrag oder die in 2.2.3 skizzierten Mischformen erlauben vor allem Improvisation des Vortrags und Interaktion mit den Lernern.[9] Gerade dann, wenn Vorlesungsaufzeichnungen in didaktischen Szenarien eingesetzt werden, in denen sie primär zur Wissensvermittlung und nicht zur Wiederholung des Gelernten dienen (vgl. 3.3 und 3.4), ist der Betrachter der Aufzeichnung allerdings ohnehin von der Interaktion ausgeschlossen. Zudem ist in derartigen Szenarien eine hohe Qualität des Materials wünschenswert, so sind beispielsweise saisonal oder kontext-bedingte Kommentare des Dozenten aus der Aufzeichnung heraus zu schneiden [Mertens, Ickerott, Witte & Vornberger 2005]. Nach [Knight, Steinbach & White 2002] sollte in aufzuzeichnenden Lehrveranstaltungen Interaktion möglichst vermieden werden bzw. Interaktion ist auf elektronische Kommunikationsmedien auszulagern. Dieser Vorschlag erscheint vor allem unter der Perspektive der Nutzung des aufgezeichneten Materials über mehrere Semester hinweg und in unterschiedlichen Kursen sinnvoll. Ein wichtiger Vorteil dieses Vorgehens ist die Tatsache, dass sich die Studierenden aktiv an der Kommunikation beteiligen können. Die Möglichkeit interaktiver Lehrstile in Tafelvorträgen, verwandten Präsentationsformen und den in 2.2.3 dargestellten Mischformen ist in Bezug auf Vorlesungsaufzeichnungen daher nicht in jedem Fall als vorteilig zu betrachten. Hinzu kommen technische Schwierigkeiten bei der Aufzeichnung derartiger Präsentationsformen wie beispielsweise die Schriftqualität in der Aufzeichnung, deren Lösungen meist mit erheblichen Mehrkosten und mit oft noch zu treibendem Forschungsaufwand verbunden sind [Lauer & Ottmann 2002].
Dem gegenüber stehen, die entsprechende Sachkenntnis im Umgang mit PowerPoint und das didaktische Wissen des Dozenten vorausgesetzt, die Möglichkeit zur Erstellung multimedialer Präsentationen, eine gute Lesbarkeit des Folienmaterials und eine Vorstrukturierung der Aufzeichnung bei Verwendung von PowerPoint. Insbesondere ist hier auch der didaktische und aufmerksamkeitspsychologische Vorteil von Animationen und Hervorhebungseffekten zu nennen.
Als Fazit lässt sich festhalten, dass die Lehrform des Tafelvortrags und verwandte Präsentationsformen für die Hochschullehre in vielen Fällen sehr gut geeignet sind, dass jedoch in den meisten Anwendungsfällen von Vorlesungsaufzeichnungen, gerade in Hinblick auf

[9] Wie in Fußnote 4 diskutiert, sind derartige Lehrformen jedoch nicht in allen Fällen zu berücksichtigen.

deren spätere Verwendung in unterschiedlichen Lernszenarien, Vorträge mit elektronischen Präsentationsmitteln und insbesondere PowerPoint besser geeignet sind.

## 2.3 Video- und audiobasierte Ansätze

In vielen existierenden Systemen zur Aufzeichnung von Vorlesungen wird der Dozent per Video aufgezeichnet. Eine Reihe weiterer Systeme bieten lediglich einen Audiomitschnitt des Vortrags. Für beide Vorgehensweisen existieren Argumente, die jedoch nicht in jedem Fall auf das aufzuzeichnende Präsentationsformat (vgl. 2.2) anwendbar sind. Im Folgenden werden zunächst die wichtigsten Argumente beider Seiten dargestellt und in Hinblick auf die im Rahmen dieser Arbeit behandelte Aufzeichnung von PowerPoint-basierten Vorlesungen analysiert. Auf dieser Analyse aufbauend wird anschließend die Entscheidung zur Implementation eines videobasierten Aufzeichnungssystems im Rahmen dieser Arbeit begründet. Eine eingehendere Betrachtung der in 2.3.1 diskutierten Split Attention Problematik bei Vorlesungsaufzeichnungen ist in [Mertens, Friedland & Krüger 2006] zu finden.

### 2.3.1 Gründe für die Verwendung rein audiobasierter Ansätze

Argumente gegen die Verwendung einer Videoaufzeichnung sind die Dateigröße des Videos, der Zusatzaufwand, der durch die Videoaufzeichnung entsteht und das Split bzw. Divided Attention Problem [Friedland 2004]. Das Split Attention Problem tritt auf, wenn Lernenden simultan Informationen visuell an verschiedenen Orten präsentiert werden und sich diese Informationen aufeinander beziehen. [Moreno & Mayer 2000] haben diesen Sachverhalt bei der gleichzeitigen Darbietung von Animationen und erläuterndem Text untersucht. In diesem Fall müssen die Lerner ihre Aufmerksamkeit zwischen zwei Medien teilen, was zu einer Verminderung der Lernleistung führt [Moreno & Mayer 2000; Brünken & Leutner 2001]. Werden zueinander passende Informationen dagegen gleichzeitig auditiv und visuell dargeboten, kann die kognitive Belastung der Lernenden verringert werden [Brünken & Leutner 2001] und eine Steigerung der Lerneffizienz erreicht werden [Mayer 2001], da die kognitive Verarbeitung der präsentierten Information durch Kongruenzeffekte der auditiven und visuellen Informationsverarbeitung erleichtert wird [Larsen, McIlhagga, Baert & Bundesen 2003]
Mit der konstant steigenden Verbreitung von Breitband-Internet-Verbindungen stellt die Dateigröße von Videoaufzeichnungen beim Einsatz von Vorlesungsmitschnitten kaum noch ein Problem dar [Dolzani & Ronchetti 2005]. Der mit einer Videoaufzeichnung verbundene Zusatzaufwand kann durch Integration der Aufzeichnungstechnik in eine vorhandene Hörsaalausstattung minimiert werden [Mertens, Knaden, Thelen & Vornberger 2005b]. Auch für das Split Attention Problem existiert eine technische Lösung, die im Rahmen des Projektes E-Kreide entwickelt wurde [Friedland 2004]. In dem dort vorgestellten Ansatz werden zur Wiedergabe aufgezeichneter Tafelvorträge zwei Videospuren so zusammengeführt, dass der Eindruck entsteht, der Dozent würde halbtransparent vor der Tafel stehen. In [Gleicher, Heck & Wallick 2002] wird ein ähnlicher Ansatz vorgestellt. Für Aufzeichnungen, die auf Folienvorträgen basieren, entsteht das Split Attention Problem erst gar nicht, da Folien und Video des Dozenten nicht gleichzeitig die Aufmerksamkeit der Studierenden beanspruchen und da Folien in der Regel keinen Fließtext, sondern Stichworte oder Grafiken enthalten, die den Vortrag unterstützen aber nicht ersetzen sollen [Mertens, Friedland & Krüger 2006].

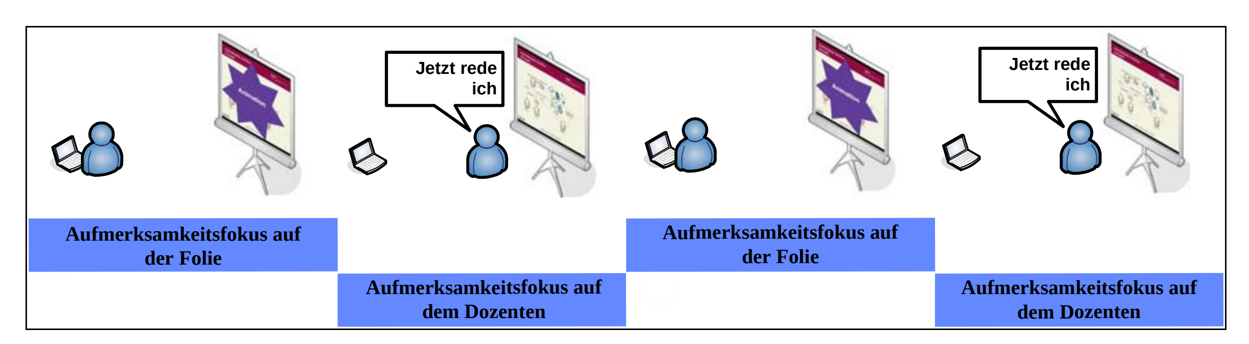



**Abb. 2-3: Wechselnder Aufmerksamkeitsfokus bei Folienvorträgen**

Abb. 2-3 stellt schematisch den Wechsel des Aufmerksamkeitsfokus bei einem Folienvortrag dar. Wenn der Dozent redet, findet in der Regel keine Folienanimation statt und wenn eine Folienanimation stattfindet, macht der Vortragende meist eine kurze Sprechpause.[10]
Hinzu kommt die Tatsache, dass die Verbindung zwischen Medium und Dozent bei einem Folienvortrag deutlich lockerer ist als bei einem Tafelvortrag, in dem der Tafelanschrieb durch die Interaktion des Dozenten mit der Tafel entsteht. Bei einem aufgezeichneten Folienvortrag sind Dozentenvideo und Folie zwar räumlich voneinander getrennt, im Gegensatz zu einer Tafelvorlesung ist diese Trennung allerdings auch in der Präsenz-vorlesung gegeben. Für die im Rahmen dieser Arbeit umgesetzte Aufzeichnung von Folienvorträgen ist das Split Attention Problem daher nicht relevant.

### 2.3.2 Gründe für die Verwendung videobasierter Ansätze

Für die Einbindung einer Videoaufnahme des Dozenten in die Aufzeichnung folienbasierter Vorlesungsaufzeichnungen sprechen eine Reihe von pragmatischen Erwägungen und Resultaten empirischer Untersuchungen.
Pragmatische Argumente für die Einbindung einer Videoaufzeichnung sind beispielsweise die Tatsachen, dass die Aufzeichnung von Studierenden durch das Video als lebendiger und authentischer empfunden wird oder dass eine Aufzeichnung mit Video mehr Information zur Verfügung stellt als eine audiobasierte Aufzeichnung [Glowalla 2004]. Beispiele hierfür sind die Körpersprache und Mimik des Dozenten [Knight, Steinbach & White 2002]. Hinzu kommt, dass die Videoaufzeichnung den Vortrag in einen situativen Kontext einbettet und evtl. in der Aufzeichnung zu erkennende Studierende einen deutlich dynamischeren und realistischeren Eindruck vermitteln.
Der Nutzen der Anreicherung einer Vorlesungsaufzeichnungen durch einen Videomitschnitt ist allerdings nicht nur von Praktikern postuliert worden, sondern wie bereits einleitend erwähnt auch in einer Reihe empirischer Studien untersucht worden. So hat beispielsweise eine Studie zum Navigationsverhalten von Nutzern im Umgang mit dem System ePresence (vgl. 5.4) gezeigt, dass die Videoaufnahme wertvolle nonverbale Hinweise wie Gestik und Mimik des Vortragenden enthalten kann [Dufour, Toms, Lewis & Baecker 2005]. [Kelly & Goldsmith 2004] haben in einer Metastudie herausgearbeitet, dass die Wiedergabe der Gestik des Dozenten einen positiven Einfluss auf die Lernleistung oder in anderen Fällen die Lernmotivation der Versuchspersonen in den durchgeführten Experimenten hatte. [Fey 2002] hat in einer Vergleichsstudie zwischen audio- und videobasierten Aufzeichnungen einen positiven Einfluss des Dozentenvideos auf die Lernmotivation von Studierenden nachgewiesen. Ähnliche Ergebnisse sind auch in einer Akzeptanzstudie von [Glowalla 2004] zu Tage getreten. In der von Glowalla durchgeführten Studie haben Studierende eine Aufzeichnung mit Video der gleichen Aufzeichnung ohne Video und der zugehörigen

[10] In [Bianchi 2004] wird ein Ansatz zum automatisierten Kameramanagement bei der Aufzeichnung von Folienvorträgen vorgestellt, in dem der hier beschriebene Wechsel zwischen diesen beiden Aufmerksamkeits-schwerpunkten erfolgreich zur Perspektivenauswahl eingesetzt wird.

Präsenzveranstaltung mit dem Argument, sie würden dort weniger abgelenkt, vorgezogen. Diese Angaben werden von einer Eye-Tracker-Untersuchung untermauert, in der die Studierenden bei einer videobasierten Aufzeichnung 70% der Nutzungszeit das Dozentenvideo visuell fixiert haben und 20 % der Zeit die Folien angeschaut haben. In der rein audiobasierten Vergleichsbedingung haben die Studierenden die Folien 60% der Nutzungszeit visuell fixiert [Glowalla 2004]. Die videobasierte Aufzeichnung hat die visuelle Aufmerksamkeit der Probanden in diesem Fall damit zu 90% der Nutzungszeit auf sich gezogen, das audiobasierte Gegenstück konnte die visuelle Aufmerksamkeit der Studierenden nur zu 60% der Nutzungszeit fesseln.

### 2.3.3 Fazit

Es existiert eine Reihe von Argumenten sowohl für als auch gegen die Einbindung eines Dozentenvideos in Vorlesungsaufzeichnungen.
In der Literatur angeführte Argumente gegen die Einbindung eines Videos sind in 2.3.1 eingehender betrachtet worden. Dabei ist die Schlussfolgerung erarbeitet worden, dass keines der bekannten Argumente sinnvoll auf die im Rahmen dieser Arbeit angestrebte Implementation anwendbar ist. Das Split Attention Problem tritt bei Folienvorträgen nicht auf, die Übertragungsgröße von Videodateien stellt gerade bei Verwendung von Streaming-Formaten immer weniger ein Problem dar und technisch/organisatorische Hürden wie der Auf- und Abbau von Videoaufzeichnungstechnik lassen sich durch eine Festinstallation der entsprechenden Technik vergleichsweise kostengünstig umgehen.
Dem gegenüber steht, wie in 2.3.2 gezeigt, eine Vielzahl von Gründen, die für die Einbettung einer synchronisierten Videoaufzeichnung des Dozenten in die Wiedergabekomponente von Vorlesungsaufzeichnungen sprechen. So wirkt sich eine Videoaufnahme positiv auf die Lernmotivation der Nutzer aus und bindet deren visuelle Aufmerksamkeit deutlich stärker als eine Aufzeichnung ohne Video. Auch können durch die Videoaufnahme nonverbale Hinweise wie Gestik und Mimik des Dozenten transportiert werden.
Eine Videoaufnahme bringt damit einen deutlichen Mehrwert im Vergleich zu einer reinen Audioaufzeichnung. Im Rahmen der vorliegenden Arbeit wird daher ein Aufzeichnungssystem implementiert, das eine Videoaufzeichnung des Dozenten mit dem genutzten Folienmaterial verbindet.

# 3 Einsatz- und Nutzungsszenarien für Vorlesungsaufzeichnungen im regulären Lehrbetrieb

Hauptargument zum Einsatz und zur Förderung von eLearning und den damit verbundenen Technologien von der Lernplattform bis zum Vorlesungsaufzeichnungssystem ist die immer wiederkehrende Forderung nach didaktischem oder ökonomischem Mehrwert. [Stockmann & Schäffer 2002] stellen ausgehend von dieser Forderung eine Konzeption zur Evaluation von E-Learning Angeboten vor, die Mehrwert-Aspekte in den Mittelpunkt der Bewertung stellt. Didaktischer Mehrwert lässt sich jedoch nicht durch Technologieeinsatz allein erreichen, sondern bedarf auch eines zugrunde liegenden didaktischen Konzeptes [Jantke 2005]. Diese Feststellung gilt in gleichem Maße auch für den ökonomischen Mehrwert von Vorlesungsaufzeichnungen. Mit der Bereitstellung einer Vorlesungsaufzeichnung im Internet oder auf einem Offline-Medium alleine ist lediglich eine Grundlage für potentielle Effizienzsteigerungen geschaffen. Darüber hinaus ist es nötig, Szenarien zu entwickeln, in denen Lehrtätigkeit durch die Verwendung von Vorlesungsaufzeichnungen unterstützt oder sogar ersetzt werden kann. Wichtig sind dabei vor allem die Akzeptanz seitens der Studierenden und deren Lernerfolg. Im weiteren Verlauf dieses Kapitels werden zunächst verschiedene identifizierbare Grundtypen von Szenarien beschrieben und hinsichtlich der Faktoren Mediendidaktik, technische Anforderungen, organisatorische Rahmenbedingungen, Nutzenpotentiale, Akzeptanz seitens der Studierenden und Anforderungen an die Software zur Wiedergabe der Aufzeichnung analysiert. Die zur Darstellung der Szenarien vorgenommene ontologische Kategorisierung ist dabei nicht als disjunkte Unterteilung sondern vielmehr als grobe Charakterisierung prototypischer Grundformen zu verstehen. Abwandlungen der beschriebenen Szenarien und Mischformen sind durchaus möglich.

Ziel der Betrachtung der einzelnen Szenarien ist in erster Linie die Erarbeitung erfolgskritischer Faktoren und die Darstellung von Möglichkeiten zur sinnvollen Einbindung von Vorlesungsaufzeichnungen in den universitären Regelbetrieb.

Darüber hinaus dient das vorliegende Kapitel dazu, Anforderungen an die verwendete Aufzeichnungs- und Wiedergabesoftware zu erarbeiten und aufzuzeigen. Besondere Relevanz gewinnt dieser Punkt in Anbetracht der Tatsache, dass auch ausgeklügelte didaktische Szenarien nur dann sinnvoll durchgeführt werden können, wenn sie von der eingesetzten Technik in ausreichendem Maße unterstützt werden.

Einige der in diesem Kapitel dargestellten Beobachtungen und Schlussfolgerungen wurden in kürzerer Form bereits in [Mertens, Knaden, Krüger & Vornberger 2004] und [Mertens, Krüger & Vornberger 2004] veröffentlicht. Ein kurzer und recht abstrakter Literaturüberblick zu pädagogischen und mediengestalterischen Fragen im weiter gefassten Kontext von Vortragsaufzeichnungen lässt sich in [Krüger 2005] finden.

## 3.1 Aufzeichnungen als ergänzendes Material

Der Einsatz von Vorlesungsaufzeichnungen als ergänzendes Material wurde in Deutschland bereits in einer Reihe von Forschungsprojekten praktiziert [Krüger 2005] und ist wahrscheinlich die am weitesten verbreitete Form der Nutzung von Vorlesungsaufzeichnungen. Aufgrund des ergänzenden Charakters des Angebots ist dieses Szenario kaum mit Risiken behaftet und eignet sich daher auch sehr gut zur Erprobung neuer Aufzeichnungssysteme oder Technologien. Wird eine einzelne Sitzung nicht aufgezeichnet oder ist eine Aufzeichnung aus technischen Gründen unbrauchbar, entstehen keine Lücken im Lehrbetrieb.

Das Szenario besteht im Wesentlichen daraus, dass zu einer im normalen Lehrbetrieb stattfindenden Vorlesung ein Video- und Audiomitschnitt mit den dazugehörigen Folien bzw.

Tafelbildern zur Verfügung gestellt wird. Studierende können die Aufzeichnung dann zu einem beliebigen Zeitpunkt nach der jeweiligen Sitzung von geeigneten Einzelplatzrechnern aus abrufen, wie schematisch in Abb. 3-1 dargestellt.

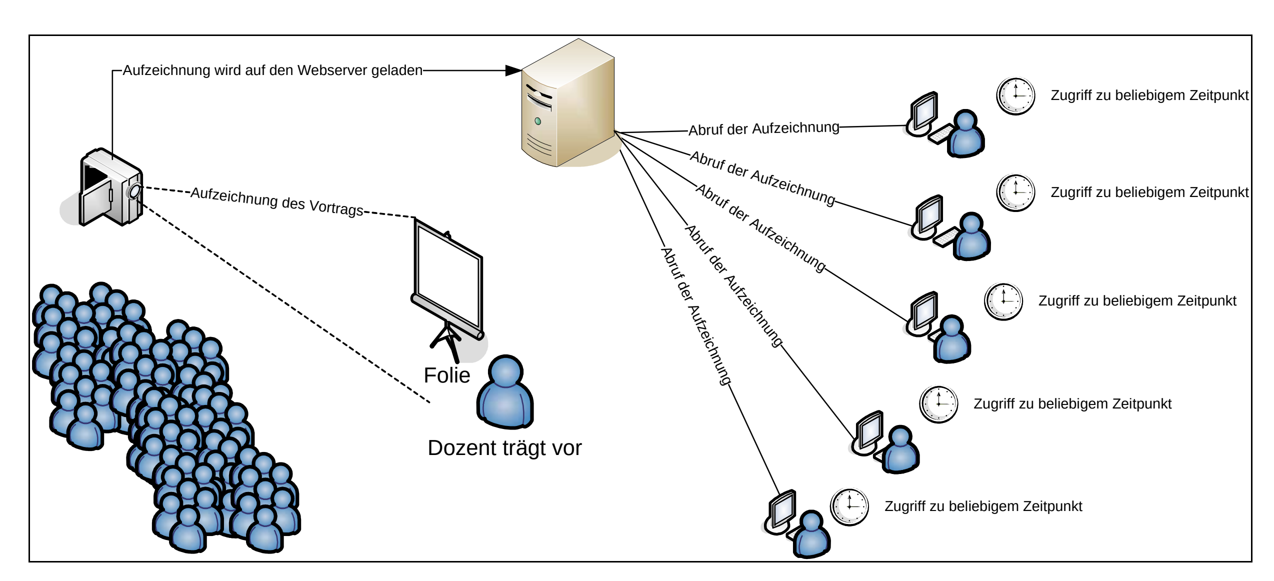



**Abb. 3-1: Schematische Darstellung des Einsatzes von Vorlesungsaufzeichnungen als ergänzendem Material**

Auch wenn dieses Einsatzszenario prinzipiell ohne allzu großen Aufwand zu implementieren ist, kann die Effektivität und der Erfolg des Szenarios massiv durch technische und organisatorische Rahmenbedingungen beeinflusst werden. In den folgenden Punkten werden daher kurz zur Durchführung des Szenarios relevante Aspekte umrissen.

### 3.1.1 Mediendidaktische Konzeption

Aufgrund des reinen Zusatzcharakters der Aufzeichnung wird das der Veranstaltung zugrunde liegende didaktische Konzept von diesem Einsatzszenario nur in einem sehr geringen Maße berührt. In verschiedenen Berichten wie z.B. [Zupancic & Horz 2002] und [Jarchow, Angilletta, Bönkost, Drawe, Ogurol, Salm & Wischnewsky 2002] wird die erfolgreiche Durchführung des Einsatzes von Vorlesungsaufzeichnungen als ergänzendes Material beschrieben. Auch an der Universität Osnabrück sind mit diesem Szenario gute Erfahrungen gemacht worden. Als wichtiges Kriterium für den Erfolg des Szenarios hat sich die Begleitung der Veranstaltung durch Übungen und Tutorien gezeigt. Diese Form der Betreuung bietet Studierenden die Möglichkeit, das in der Vorlesung Gelernte zeitnah anzuwenden und zu rekapitulieren. Die Bedeutung derartiger Begleitveranstaltungen ist dabei in diesem Szenario sogar noch deutlich höher als in vergleichbaren reinen Präsenzveranstaltungen, da die Studierenden so zur zeitnahen Nacharbeitung des Vorlesungsstoffes motiviert werden können. Darüber hinaus ist die Anreicherung des aufgezeichneten Materials durch begleitende asynchrone Lernunterstützungsmittel wie Online-Diskussionsforen oder Materialien zur Selbstlernkontrolle sinnvoll [Jarchow, Angilletta, Bönkost, Drawe, Ogurol, Salm & Wischnewsky 2002].
Als didaktisch ungeeigneten Ansatz schildern [Bell, Cockburn, McKenzie & Vargo 2001] einen Versuch, in dem Vorlesungsaufzeichnungen als ergänzendes Material zu einer Vorlesung im Grundstudium ohne Übungen und Tutorien angeboten wurden. In diesem Kontext hat die Existenz der Aufzeichnung viele der Veranstaltungsteilnehmer zu der Fehlannahme verleitet, den gesamten Stoff jederzeit, also auch erst kurz vor der Klausur lernen zu können [Bell, Cockburn, McKenzie & Vargo 2001]. In Vorlesungen, die ohne begleitende Tutorien oder Übungen angeboten werden, ist der Einsatz von Vorlesungsauf-

zeichnungen als ergänzendes Material daher nur dann Erfolg versprechend, wenn die Studierenden ein ausreichendes Maß an Eigenmotivation und Studienerfahrung mitbringen.

### 3.1.2 Technische Anforderungen

Die technischen Anforderungen dieses Szenarios lassen sich grob in die Bereiche Aufzeichnungs- und Produktionstechnik, Datenhaltung und Bereitstellung des aufgezeichneten Materials sowie technische Infrastruktur zur Nutzung durch die Studierenden unterteilen.
Die Anforderungen an die zu verwendende Aufnahmetechnik beschränken sich auf die Möglichkeit zur Audio- und Videoaufzeichnung des Dozenten und einen Rechner zur Digitalisierung der Aufzeichnung. Eventuell ist die Möglichkeit eines Livestreams zu berücksichtigen, die eine synchrone Übertragung während der Aufzeichnung erlaubt. Bei Einbettung des Szenarios in einen videokonferenzgestützen Veranstaltungsaustausch kann bereits vorhandene Technik wie für die Übertragung eingesetzte Kameras und Tontechnik mitgenutzt werden. Hierbei ist jedoch ein Aufbau sinnvoll, bei dem die Auswirkungen von Übertragungsstörungen auf die Aufzeichnung minimiert werden. Wird eine Vorlesung von einem Standort aus übertragen, sollte wenn möglich an diesem Standort aufgezeichnet werden. D. h., dass nicht nur Kamera und Tontechnik sondern auch die zur Aufzeichnung verwendeten Digitalisierungsrechner oder Videorekorder am Aufzeichnungsstandort vorhanden sein sollten. Ist dies nicht machbar, sollten ggf. niedrigere Übertragungsraten in Kauf genommen werden, um Verbindungsabbrüche zu vermeiden.
Zur Datenhaltung und Bereitstellung des aufgezeichneten Materials ist ein Webserver zur Bereitstellung von Folien und verwendeten Dokumente sowie ein Streamingserver für das verwendete Video- oder Audioformat erforderlich. Ist kein solcher Server vorhanden oder das verwendete Format nicht streamingfähig, können auch Audio- oder Videodateien auf einem herkömmlichen Webserver zum Download bereitgestellt werden. Bei derartigen Downloadmöglichkeiten sollte im Vorfeld das Einverständnis der Dozenten der aufgezeichneten Veranstaltungen eingeholt, um deren Persönlichkeitsrechte zu wahren. Eine ausschließliche Bereitstellung der Aufzeichnung auf Offline-Medien erscheint angesichts der in [Bell, Cockburn, McKenzie & Vargo 2001] beschriebenen Erfahrungen nicht sinnvoll.
Für die Studierenden, die beim Zugriff auf die Nutzung der Rechnerausbildungsräume der Hochschule angewiesen sind, ist ggf. sicherzustellen, dass an den dortigen Rechnerarbeitsplätzen die zum Abspielen des Materials erforderliche Software installiert ist. Eine Umfrage an der Universität Bremen im Jahr 2002 hat in diesem Zusammenhang gezeigt, dass dort 38% der befragten Studierenden universitätseigene CIP-Räume und 48% Rechner in den Räumen der Universitätsbibliothek nutzen [Jarchow, Angilletta, Bönkost, Drawe, Ogurol, Salm & Wischnewsky 2002]. In Anbetracht der immer weiter fortschreitenden Verbreitung von WLan-fähigen Laptops ist auch die Unterstützung des Empfangs von Vorlesungsaufzeichnungen über universitätseigene Funknetze sinnvoll. Wenn Teile der Aufzeichnung in begleitenden Tutorien oder Übungen verwendet werden sollen, sind entsprechende Medienräume bereitzustellen.

### 3.1.3 Organisatorische Rahmenbedingungen

Aufgrund des Zusatzcharakters des Angebotes bedingt der Einsatz von Vorlesungsaufzeichnungen als ergänzendes Material keine wesentlichen Änderungen des Lehrbetriebes. Lediglich die in 3.1.1 dargestellte teilweise Notwendigkeit des Angebots von begleitenden Übungen oder Tutorien ist hinsichtlich der dort erörterten Problematik mangelnder konstanter Motivation der Studierenden zu berücksichtigen.
Darüber hinaus ist die technische Betreuung der Aufzeichnung und die Bereitstellung der aufgezeichneten Materialien zu organisieren. Gegebenenfalls kommt auch eine inhaltliche Nachbearbeitung der Aufzeichnung wie das Herausschneiden bestimmter Passagen hinzu. In

der Regel sollte eine derartige Nachbearbeitung jedoch nicht notwendig sein. Die technische Betreuung der Aufzeichnung umfasst je nach Ausstattung der verwendeten Hörsäle neben Kameraführung auch Auf- und Abbau sowie Transport der benötigten Technik. Abhängig von der verwendeten technischen und administrativen Infrastruktur lässt sich dieser Aufwand jedoch faktisch auf Null reduzieren [Mertens, Knaden, Thelen & Vornberger 2005a].

### 3.1.4 Nutzenpotentiale

Ein hohes Nutzenpotential ist besonders in thematisch hoch verdichteten Lehrveranstaltungen gegeben, in denen komplexes Wissen in kurzer Zeit vermittelt wird. Durch den Einsatz moderner Präsentationsmedien wie beispielsweise PowerPoint können Lehrende weitaus mehr Information in sehr kurzer Zeit präsentieren als mit herkömmlichen Präsentationsmedien [Brotherton 2001] möglich.[11] Auch in Lehrveranstaltungen mit hohem formalwissenschaftlichem Anteil wie beispielsweise in der Mathematik lässt sich eine sehr hohe Informationsdichte oft nicht vermeiden. Dieser Sachverhalt führt in herkömmlichen Lehrveranstaltungen, die nicht durch Aufzeichnungen unterstützt werden oft dazu, dass die Teilnehmer der Veranstaltung in zu hohem Maße mit der Mitschrift der Vorlesung befasst sind und sich so nicht hinreichend auf den vermittelten Lehrstoff konzentrieren können [Brotherton 2001].

Wird eine Veranstaltung aufgezeichnet, entfällt die Notwendigkeit des Mitschreibens während der Vorlesung und die Studierenden können sich besser auf das vermittelte Wissen konzentrieren. Darüber hinaus können kritische oder nicht verstandene Passagen des Vortrags beliebig oft wiederholt werden. Dieser Faktor kann sich nicht nur bei thematisch schwierigen Punkten, sondern auch bei sprachlichen Schwierigkeiten seitens der Studierenden als hilfreich erweisen [Krüger, Klie, Heinrich & Jobmann 2005]. Eine Studie von [Zupancic & Horz 2002] zeigt, dass Vorlesungsaufzeichnungen von Studierenden auch in hohem Maße zur Vorbereitung von Prüfungen und Klausuren verwendet werden. Weitere Nutzenpotentiale sind gegeben, wenn Studierende eine Sitzung der Veranstaltung versäumen oder eine zeitlich parallel liegende Veranstaltung besuchen.

### 3.1.5 Akzeptanz seitens der Studierenden

Durch den add-on-Charakter des Angebotes können Studierende die in 3.1.4 dargestellten Nutzenpotentiale des Szenarios ohne Einbußen jeglicher Art in Anspruch nehmen. Der in 3.1.1 näher erläuterte didaktische Ansatz des Szenarios kommt Studierenden durch ein erhöhtes Betreuungsangebot entgegen. Das Szenario bietet damit eine Vielzahl von Annehmlichkeiten ohne die Studierenden in irgendeiner Hinsicht mehr zu fordern als eine vergleichbare herkömmliche Veranstaltung ohne Aufzeichnung. Als weiterer die Akzeptanz fördernder Faktor kann die Tatsache gewertet werden, dass der gewohnte Veranstaltungs-ablauf durch das zusätzliche Angebot nicht verändert wird.

Auf Seiten der Studierenden findet dieses Szenario daher nach verschiedenen Berichten durchgehend positive Resonanz [Zupancic & Horz 2002; Krüger, Klie, Heinrich & Jobmann 2005; Brotherton 2001]. Auch am Standort Osnabrück wurden mit diesem Szenario bezüglich der Akzeptanz seitens der Studierenden durchweg positive Erfahrungen gemacht.

### 3.1.6 Wiedergabemöglichkeiten

Die Anforderungen an das Wiedergabemedium der Aufzeichnung variieren mit den zu erzielenden Nutzenpotentialen (vgl. 3.1.4). So ist beispielsweise ein linearer Datenstrom

[11] Wenn es sich bei dieser Information beispielsweise um ergänzendes Kontextwissen handelt, kann dadurch auch ein wesentlicher didaktischer Nutzen entstehen [Brotherton 2001].

ohne Navigationsmöglichkeiten zum Nachholen einer verpassten Sitzung der Veranstaltung durchaus ausreichend, wenn auch nur mäßig komfortabel. Soll die Aufzeichnung im Gegensatz dazu ähnlich wie ein Vorlesungsskript genutzt werden, müssen Möglichkeiten zur feinkörnigen Navigation innerhalb der Aufzeichnung gegeben sein [Hürst 2001]. Auch für die in [Krüger, Klie, Heinrich & Jobmann 2005] und [Lanham & Zhou 2004] dargestellte Nutzung bei sprachlichen Verständnisschwierigkeiten sind weiterführende Navigationsmöglichkeiten essentiell.

Zur Nutzung und Diskussion der Aufzeichnung in herkömmlichen Tutorien oder auch Online-Diskussionsforen muss die Möglichkeit bestehen, einzelne Passagen zu referenzieren und Schlüsselpositionen direkt abzuspielen. Die Möglichkeit, einzelne Passagen zu referenzieren bietet dabei besonders in der Online-Kommunikation einen Weg, den Gegenstand der Diskussion eindeutig festzulegen. Darüber hinaus sind generelle Qualitätsmerkmale wie beispielsweise die Darstellungsqualität der gezeigten Folien [Ziewer & Seidl 2002] oder auch die Qualität der mitgeschnittenen Tonspur [Baecker 2003] zu beachten.

## 3.2 Aufzeichnungen als zeitfix präsentierter Vorlesungsersatz

Die Vorführung einer Vorlesungsaufzeichnung als zeitfix präsentierter Vorlesungsersatz ist anschaulich mit einer Kinovorführung der Aufzeichnung vergleichbar. Zu regelmäßigen Terminen wird beispielsweise ein- oder zweimal wöchentlich in einem technisch geeigneten Hörsaal die Aufzeichnung der Vorlesung auf großer Leinwand gezeigt. Die Aufzeichnung stammt dabei normalerweise aus einem vorangegangenen Semester. Wird das Szenario durchgeführt, um Terminkollisionen bei zeitlich parallel liegenden Veranstaltungen zu vermeiden, kann die Aufzeichnung auch im jeweils aktuellen Semester entstanden sein. Die Vorführung ersetzt dabei vollständig den Vortrag des Lehrenden. Der Lehrende muss dabei nicht zwingend anwesend sein. Zur Diskussion von eventuell auftretenden Fragen sollte jedoch ein Übungsleiter oder Tutor anwesend sein.

Das Szenario selbst ist sowohl im Aufbau als auch in der Konzeption sehr eng an eine konventionelle Vorlesung angelehnt, wie auch die schematische Darstellung in Abb. 3-2 erkennen lässt. Der entscheidende Unterschied besteht in der Substitution des Präsenzvortrags durch eine Videoaufzeichnung.

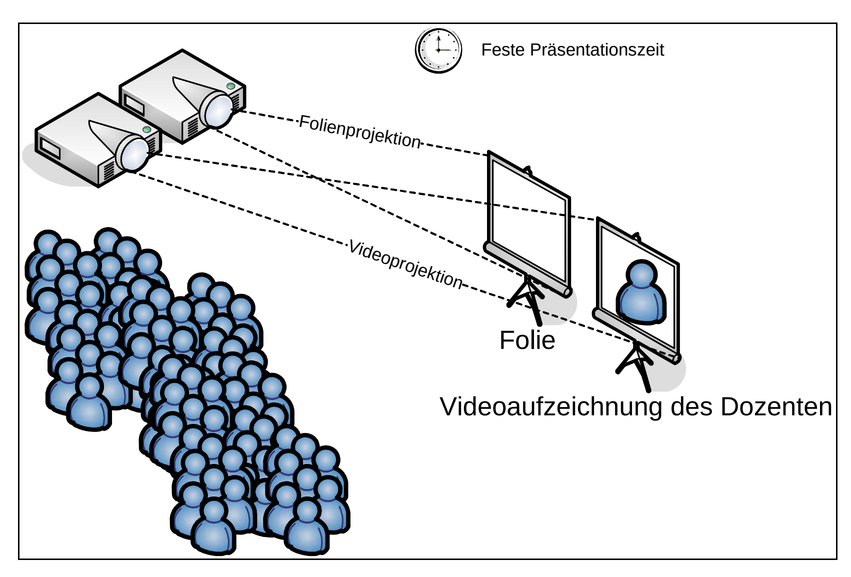



**Abb. 3-2: Schematische Darstellung der Präsentation von Mitschnitten als zeitfix präsentierter Vorlesungsersatz**

Trotz der großen Ähnlichkeit zu herkömmlichen Vorlesungen in einigen Bereichen entsteht durch das Fehlen des Vortragenden als reale Person in der Wahrnehmung der Studierenden

eine völlig neue Situation. Zur effektiven Durchführung des Szenarios sind daher nicht nur in technischer Hinsicht, sondern auch im Hinblick auf die Einführung, Organisation und Betreuung des Szenarios eine Reihe erfolgskritischer Faktoren zu berücksichtigen, die im weiteren Verlauf dieses Abschnittes eingehender diskutiert werden.

### 3.2.1 Mediendidaktische Konzeption

Bei der didaktischen Konzeption des Einsatzes von Vorlesungsaufzeichnungen als zeitfix präsentierter Vorlesungsersatz ist in erster Linie der bereits dargestellte Wegfall der direkten Präsenz des Dozenten beim Vortrag zu berücksichtigen. In einer herkömmlichen Vorlesung übt der Dozent durch seine Anwesenheit und durch die Möglichkeit, direkt mit den Zuhörern zu interagieren direkten Einfluss auf gruppendynamische Prozesse und auch auf das Verhalten einzelner Studierender aus. So kann, wie beispielsweise in einem experimentellen Einsatz des Szenarios an der Universität Osnabrück beobachtet, ein Anstieg des Schwierigkeitsgrades während der Vorlesung sehr leicht dazu führen, dass die Studierenden, die dem Vortrag nicht mehr folgen können, den Hörsaal verlassen, um ihre Zeit subjektiv und auch objektiv betrachtet besser zu nutzen. So hat eine anschließende Auswertung von Klausurergebnissen im Vergleich mit Klausurergebnissen der Vorjahre und auch im Vergleich mit dem in 3.3 beschriebenen zeitvariablen Einsatzszenario eine signifikant höhere Erfolgsquote gezeigt.[12]

Darüber hinaus haben auch externe Faktoren wie beispielsweise die Öffnungszeiten der Mensa einen größeren Einfluss auf das Anwesenheitsverhalten der Studierenden, da der durch die Anwesenheit des Dozenten erzeugte soziale Druck in diesem Szenario entfällt.

Die zeitliche Positionierung der Veranstaltung sollte daher Faktoren wie die Nähe zum Wochenende, extrem frühe oder späte Anfangszeiten und auch die zeitliche Nähe zur Mittagspause berücksichtigen. Sprunghafte Anstiege der Lernkurve in einzelnen Sitzungen sollten vermieden werden.

Weitere vom didaktischen Aufbau herkömmlicher Vorlesungen abweichende Anforderungen sind nicht in besonderem Maße gegeben. Einen wesentlichen Grund hierfür stellt der wöchentliche Rhythmus der Veranstaltung dar. Er bietet den Studierenden ein festes Zeitgerüst und hält so zur kontinuierlichen Beschäftigung mit dem Lehrstoff an. Gerade in Anbetracht der durch das Szenario frei werdenden Lehrkapazität ist allerdings ein zusätzliches Angebot in Form von Übungen oder Tutorien sinnvoll.

### 3.2.2 Technische Anforderungen

Die technischen Anforderungen des Szenarios sind bezüglich der verwendeten Hardware vergleichsweise gering, da lediglich die Wiedergabe der Aufzeichnung zu bewerkstelligen ist. Um eine qualitativ hochwertige Wiedergabe zu erreichen, bietet sich die Verwendung von zwei Videoprojektoren zur gleichzeitigen Darbietung von Film und Folie [Knaden & Rolf 2003] an. Um eine synchrone Wiedergabe von Folien und Film zu ermöglichen, kann ein Rechner mit zwei Grafikkarten verwendet werden.

Die Tonspur der Aufzeichnung kann, wenn es die vorhandene Hörsaaltechnik erlaubt, direkt in die Tonanlage des Hörsaals eingespeist werden. Die notwendigen Einstellungen sind nach

[12] Eine genaue Beschreibung der im zeitvariablen Vergleichsszenario angebotenen Lehrmaterialien ist in [Mertens, Ickerott, Witte & Vornberger 2005] zu finden. Das im zeitvariablen Szenario angebotene Material war qualitativ weitaus hochwertiger als das im zeitfixen Präsentationsmodus verwendete Material, wurde jedoch nur in einer offline-Version angeboten. Das zeitfix Präsentationsszenario wurde im entsprechenden Semester durch ein online-Angebot der gezeigten Filme erweitert. Im Zusammenhang mit den in [Bell, Cockburn, McKenzie & Vargo 2001] dargestellten Erfahrungen mag dieser Sachverhalt als Indiz dafür gewertet werden, dass zeitvariabel eingesetzte Offline-Angebote sowohl als erweiternde Angebote als auch Einzelangebote von der Mehrzahl der Studierenden weniger intensiv genutzt werden.

Anleitung auch von Hilfskräften durchführbar. Dabei ist sicherzustellen, dass im Anschluss an die Aufzeichnung der Ursprungszustand wieder hergestellt wird, um Tonprobleme in nachfolgenden Veranstaltungen zu vermeiden.
Aus der Tatsache, dass die Aufzeichnung auf großer Leinwand und vor großem Publikum gezeigt wird, ergeben sich erhöhte Anforderungen an die Wiedergabequalität der verwendeten Materialien. Diese Anforderungen wirken sich nicht nur, wie in 3.2.6 dargestellt, auf die Art der zu verwendenden Medien, sondern auch auf die bei der Aufzeichnung einzusetzende Technik aus. Sowohl bei der Ton- als auch der Filmaufzeichnung sind daher höhere Qualitätsmaßstäbe anzulegen.

### 3.2.3 Organisatorische Rahmenbedingungen

Da die Veranstaltung wie eine herkömmliche Vorlesung zu festen Terminen stattfindet, ist sicherzustellen, dass die Vorführung auf Anhieb reibungslos funktioniert. Diese Anforderung ergibt sich zum einen aus der Tatsache, dass zu jedem Termin ein festgelegtes Thema behandelt wird und zum anderen daraus, dass die Studenten bei technischen Schwierigkeiten sehr schnell das Vertrauen in die Veranstaltungsform verlieren. Es ist daher wichtig, dass der technische Ablauf der Vorführung bereits frühzeitig in der vorlesungsfreien Zeit mit dem beteiligten Personal geprobt wird. Bei derartigen Probeläufen wird schnell klar, wo technische Probleme auftreten können und ob Ausrüstung fehlt. Darüber hinaus können so Arbeitsabläufe wie die Umstellung der Tonanlage oder der Anschluss des Wiedergaberechners an zwei Projektoren an beteiligte Hilfskräfte vermittelt werden.
Neben der Organisation des Ablaufs der Vorführung als Kernveranstaltung sind begleitende Übungen und Tutorien zu planen und vorzubereiten. Tutorien und Übungen können dabei in der Regel genau so durchgeführt werden, wie im regulären Lehrbetrieb. Ist ein zusätzliches Betreuungsangebot durch den Lehrenden geplant, sind hier evtl. Materialien vorzubereiten.

### 3.2.4 Nutzenpotentiale

Größere Nutzenpotentiale sind vor allem dann gegeben, wenn Lehrende nicht das gesamte Semester, bzw. nur an wechselnden Terminen über das Semester verteilt zur Verfügung stehen (Veranstaltung von Berufspraktikern, ortsfremden Dozenten, Personal, das nur teilzeitbasiert zur Lehre eingesetzt wird, krankheitsbedingte Ausfälle). Alternativ bietet sich das Verfahren an, wenn die Hörerzahl einer Veranstaltung die maximal verfügbare Teilnehmerkapazität von Hörsälen überschreitet oder die Veranstaltung je nach Studienfachkombination der Hörer mit anderen Pflichtveranstaltungen kollidiert. Dieser Punkt gewinnt vor allem mit der zunehmenden Modularisierung von Studiengängen und mit der allgemein angestrebten Verkürzung von Studiendauern an Bedeutung. Auch können Veranstaltungen häufiger angeboten werden, so dass Wartezeiten von teilweise mehreren Semestern zwischen dem Angebot thematisch gleicher Veranstaltungen entfallen können.
Im Vergleich mit anderen Szenarien stellen sich vor allem das feste Zeitgerüst und die Möglichkeit zur Nutzung der Materialien ohne besondere Hardware und ohne technische Vorkenntnisse als Vorteil heraus. Aufgrund der festen räumlichen und zeitlichen Bindung kann der Veranstaltungstyp auch im Grundstudium und in Veranstaltungen mit geringerer Studienmotivation eingesetzt werden. Neben der festen Einordnung in den wöchentlichen Stundenplan der Studierenden spielt hier sicherlich auch der Kino-Effekt des gemeinsamen Erlebens eine wichtige Rolle. Der Wegfall technischer Hürden stellt einen entscheidenden Faktor bei der Durchführbarkeit des Szenarios in technikfernen Studiengängen dar. Nichtsdestotrotz sollte den Studierenden zusätzlich die Möglichkeit zur Wiederholung nicht verstandener Passagen durch die individuelle Nutzung der Aufzeichnung an Einzelplatzrechnern gegeben werden. Zusätzliches Nutzenpotential ist durch die Möglichkeit einer interaktiveren Gestaltung der Vorführung bei Betreuung durch Tutoren gegeben (vgl. 3.2.5).

### 3.2.5 Akzeptanz seitens der Studierenden

Abgesehen vom Fehlen des Dozenten ist der Veranstaltungstyp mit einer herkömmlichen Vorlesung vergleichbar. Wie bereits in 3.2.4 dargestellt, bleiben viele von Studierenden als positiv empfundene Merkmale wie das gemeinsame Veranstaltungserleben und die Möglichkeit zur Teilnahme ohne technische Hürden erhalten. Durch diese Faktoren ist prinzipiell eine sehr hohe Akzeptanz des Szenarios durch die Studierenden gegeben. Als erfolgskritischer Faktor hat sich in diesem Zusammenhang bei Versuchen an der Universität Osnabrück die Einführung und Motivation des Szenarios seitens des oder der Lehrenden erwiesen. Besonders dann, wenn der Lehrende prinzipiell die Möglichkeit hätte, die Vorlesung persönlich zu halten, muss den Studierenden eindeutig vermittelt werden, welchen Nutzen die Veranstaltungsform für sie hat. Wenn den Studierenden klar ist, dass die Veranstaltungsform ihnen persönlich beispielsweise durch ein verbreitertes Lehrangebot oder durch erhöhte räumliche Flexibilität nutzt, ist die Akzeptanz vergleichsweise hoch. So sind an der Universität Osnabrück in den letzten Jahren sehr gute Erfahrungen mit der konzeptionell sehr ähnlichen Veranstaltungsform der Live-Übertragung von Vorlesungen gemacht worden [Knaden & Rolf 2003] [Knaden & Zettel 2002]. Dabei bestand teilweise keine Verbindung für Rückfragen zum Ausstrahlungsort, so dass sich eine Übertragbarkeit der Beobachtungen auf das beschriebene Szenario ergibt. Trotz anfänglicher technischer Probleme und der Möglichkeit jederzeit an der Live-Vorlesung teilzunehmen, verfolgten viele Studierende die Vorlesung als Übertragung.

Eine geringe Akzeptanz kann bei Studierenden vor allem durch das Gefühl hervorgerufen werden, dass der oder die Lehrende sich nur einer lästigen Pflicht entledigen will. Dieser Eindruck kann sehr leicht entstehen, wenn der Dozent oder ein wissenschaftlicher Mitarbeiter direkt nach der Anmoderation des Szenarios demonstrativ seine Unterlagen zusammenpackt und den Hörsaal verlässt. Auch der Abbau von zur Anmoderation verwendeter Technik während der Vorführung ist zu vermeiden. Bei einer Anmoderation in der ersten Veranstaltungssitzung sollte der Dozent daher die gesamte Sitzung lang anwesend sein und die Vorführung mitverfolgen.

Das in 3.2.1 beschriebene Verhalten von Studierenden, die die Veranstaltung dann verlassen, wenn die Lernkurve zu stark ansteigt oder wenn ihnen der aktuell behandelte Lehrstoff bereits bekannt ist, muss nicht zwingend als ein Indikator für niedrige Akzeptanz gewertet werden. Derartiges Verhalten kann auch als effektives Zeitmanagement verstanden werden, das im E-Learning häufig zu einer höheren Lerneffizienz führt (vgl. dazu 3.6.1).

In einigen Fällen bringen Studierende das Argument vor, dass in der Veranstaltung keine Fragen gestellt werden können. Beim Einsatz des Szenarios in Großveranstaltungen ist dieser Kritik entgegenzuhalten, dass dort aufgrund der Gruppengröße in der Regel ohnehin keine Fragen gestellt werden [Bacher & Ottmann 1995]. Interessanterweise ist dieser Kritikpunkt an der Universität Osnabrück sogar in einem Modellversuch aufgetreten, in dem die Studierenden durch ein zusätzliches Betreuungsangebot des Dozenten vermehrt die Möglichkeit bekommen haben, Fragen zu stellen. In der Aufzeichnung aus dem Vorsemester sind von den Teilnehmern der Veranstaltung nachweislich keine Fragen gestellt worden. Zur Erhöhung der Akzeptanz des Szenarios ist es daher wichtig, den Teilnehmenden das Gefühl zu geben, dass sie jederzeit Fragen stellen können. Diese Anforderung kann in der Regel dadurch abgedeckt werden, dass die technische Betreuung von fachlich vorgebildeten Hilfskräften übernommen wird, die in diesem Fall in einer Doppelrolle als technisches Personal und Tutor agieren. Da die soziale Distanz zwischen Tutoren und Studierenden geringer ist als zwischen Studierenden und Dozenten und da die Gruppengröße oft mit dem Anstieg der Lernkurve abnimmt (vgl. 3.2.1), ist in diesem Fall sogar eher mit Verständnisfragen zu rechnen als in einer herkömmlichen Vorlesung.

### 3.2.6 Wiedergabemöglichkeiten

Videoaufzeichnungen mit schlecht aufgelöster und körniger Darstellung oder groben Codierungsartefakten, die in vielen anderen Szenarien eher akzeptabel sind, führen bei einer Vorführung auf großer Leinwand sehr schnell zu einer negativen Grundhaltung gegenüber dem aufgezeichneten Material. Es entsteht der Eindruck mangelhafter Qualität, der sich schnell auch auf die Inhalte der Vorlesung und das gesamte Veranstaltungskonzept überträgt. Neben guter Tonqualität, die sich auch in einer Reihe anderer Szenarien als wichtiger und nicht in jedem Fall unproblematischer Faktor erwiesen hat [Baecker 2003; Stringer 2003], kommt in diesem Szenario daher der Qualität der Videoaufzeichnung eine besondere Rolle zu. Eine hinreichend gute Videoqualität bedarf neben der Verwendung geeigneter Aufzeichnungstechnik (vgl. 3.2.2) auch der Verwendung geeigneter Medienformate und Kompressionsstufen. Dies führt in der Regel zu vergleichsweise großen Videodateien. Auch die Darstellungsqualität der Folien ist auf Grund der Vorführung auf großer Leinwand ein kritischer Faktor. Eine Foliendarstellung in schlechter Auflösung, die auch bei anderen Einsatzszenarien von Vorlesungsaufzeichnungen negative Auswirkungen auf die Nutzbarkeit des Materials hat [Ziewer & Seidl 2002], ist hier daher unbedingt zu vermeiden. Aufgrund der sehr hohen Auflösung der Foliendarstellung durch die zu verwendenden Videobeamer ist eine vektorbasierte Darstellung ideal.

Die Verwendung von zwei Videoprojektoren, mit denen getrennt voneinander Film und Folien jeweils auf unterschiedlichen Leinwänden dargestellt werden (vgl. 3.2.2) stellt dahin gehend spezielle Anforderungen an die verwendete Wiedergabesoftware, dass beide Medienströme auf getrennten Grafikausgängen wiedergegeben werden müssen. Eine derartige Funktionalität wird nur von wenigen Systemen wie dem DVDconnector [Jantke 2005] oder dem im Rahmen dieser Arbeit entwickelten virtPresenter implementiert, ist jedoch bei Vorhandensein der entsprechenden Synchronisationsinformation sowie Film und Folien als einzeln darstellbaren Dateien unter Verwendung von Werkzeugen wie dem Real Media Editor mit Real Events recht einfach zu realisieren [Mertens, Krüger & Vornberger 2004].

Ist eine kumulative Durchführung des Szenarios in Verbindung mit anderen Szenarien wie beispielsweise der ergänzenden Bereitstellung der Aufzeichnung geplant, sind zusätzlich die jeweils dort beschriebenen Anforderungen zu berücksichtigen.

## 3.3 Aufzeichnungen als zeitvariabel präsentierter Vorlesungsersatz

Der Einsatz von Vorlesungsaufzeichnungen als zeitvariabel präsentierter Vorlesungsersatz wird in der Literatur oft als praktikables und erfolgreiches Einsatzszenario beschrieben [Effelsberg 2003; Stringer 2003; Foertsch, Moses, Strikwerda & Litzkow 2002; Hurson & Kavi 1999]. Wie in Abb. 3-3 schematisch dargestellt, entfällt die Vorlesung in diesem Szenario und wird durch die Bereitstellung einer Aufzeichnung ersetzt, die von den Studierenden zu selbstbestimmten Lernterminen durchgearbeitet wird. Die Aufzeichnung kann dabei online oder auch zusätzlich als Offline-Medium vorliegen, eine Online-Version oder zumindest die Möglichkeit zum Download der Dateien des Offline-Mediums sollte jedoch in jedem Fall gegeben sein (vgl. dazu Fußnote 12 auf Seite 40).

Das Szenario ordnet sich in den curricularen Verlauf ein und stellt somit eine vollwertige Lehrveranstaltung dar. Leistungsnachweise werden wie in einer herkömmlichen Vorlesung durch abschließende Klausuren oder in begleitenden Praktika erbracht. Eine kontinuierliche Begleitung durch Übungen oder Tutorien sollte gegeben sein, ist jedoch wie in [Stringer 2003] beschrieben auch online durch kurze wöchentliche Tests und eine Betreuung über Mail und Foren machbar.

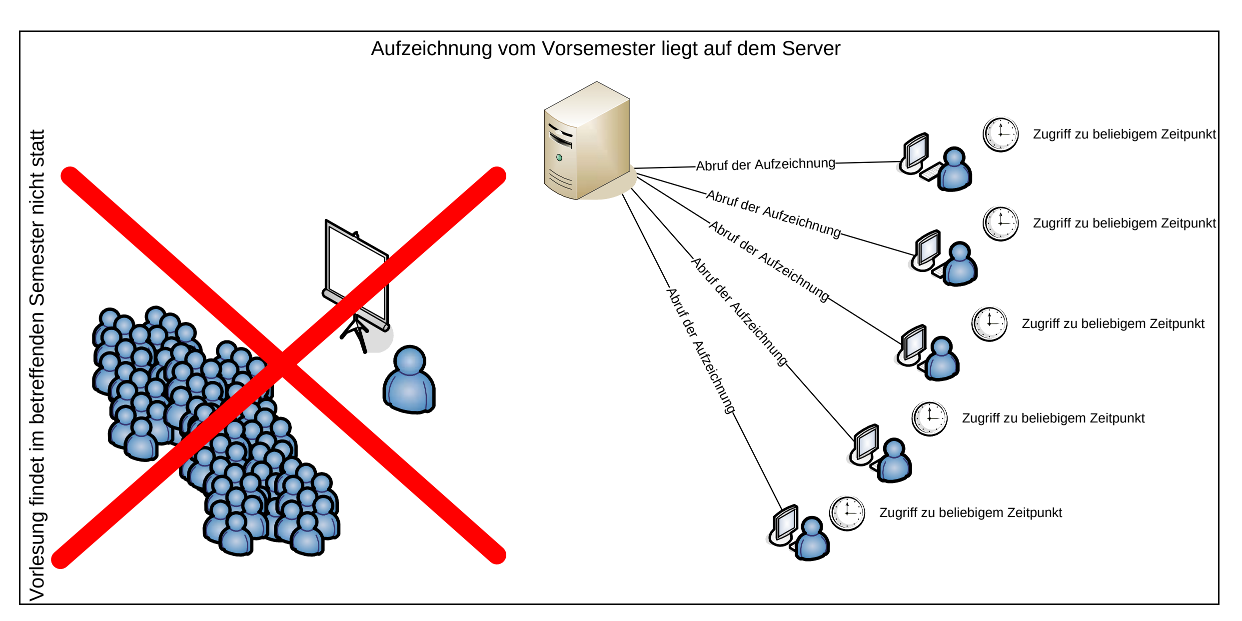



**Abb. 3-3: schematische Darstellung des Einsatzes von Mitschnitten als zeitvariabel präsentierter Vorlesungsersatz**

[Foertsch, Moses, Strikwerda & Litzkow 2002] beschreibt das Szenario als eine Umkehrung des Vorlesungs-Hausaufgaben-Paradigmas und deutet damit die Verschiebung der Rolle des Dozenten von der reinen Wissensvermittlung hin zur kollaborativen Interaktion an. Die Hauptaufgabe des Dozenten besteht nicht mehr in der Vermittlung von Wissen, da sich die Studierenden den Lehrstoff in Eigenverantwortung mit dem Medium der Vorlesungsaufzeichnung aneignen. Der Dozent leitet die Studierenden vielmehr bei der Bearbeitung von Übungsaufgaben an und klärt Fragen, die sich beim Durcharbeiten des Materials ergeben haben. In der Praxis bedeutet diese Umkehrung in vielen Punkten weitaus weniger Umstellung als die Formulierung auf Anhieb vermuten lässt. Besonders der geänderte Nutzungskontext des Materials und der durch das Szenario bedingte erhöhte Betreuungsbedarf stellen jedoch veränderte Anforderungen an die Mediengestaltung und an die Organisation der Lehrveranstaltung.

### 3.3.1 Mediendidaktische Konzeption

Die Aufgabe des Lehrenden ist in diesem Szenario ähnlich wie auch in dem in 3.2 beschriebenen Szenario nicht die Vermittlung von Wissen, sondern die Betreuung der Studierenden während des Lernens. [Foertsch, Moses, Strikwerda & Litzkow 2002] stellen hierzu ein didaktisches Konzept vor, in dem sich die tutorielle Betreuung der Studierenden in zwei wöchentliche Termine, einen Fragetermin und einen Termin zur Abgabe der Aufgaben in Dreiergruppen, unterteilt. Als zusätzliches Angebot zur Lernkontrolle, werden kurze Online-Tests bereitgestellt. Mit ähnlich intensiver Betreuung wurde das beschriebene Szenario mit einem Offline-Medium an der Universität Osnabrück durchgeführt [Mertens, Ickerott, Witte & Vornberger 2005]. [Effelsberg 2003] berichtet von der erfolgreichen Durchführung des Szenarios in Verbindung mit Präsenzübungen. [Smeaton & Keogh 1999] beschreiben den Einsatz von Vorlesungsaufzeichnungen als zeitvariablem Vorlesungsersatz in Verbindung mit regelmäßigen Präsenztutorien und Online-Übungen. Die zeitlichen Abstände zwischen den Präsenzveranstaltungen betrugen hierbei vierzehn Tage. [Stringer 2003] stellt eine Variation des Szenarios vor, in der die Betreuung der Studierenden komplett online durchgeführt wird.
Als Fazit aller in den jeweiligen Berichten dargestellten Erfahrungen lässt sich die intensive Betreuung der Studierenden in begleitenden Präsenzveranstaltungen aber auch online als ein entscheidender Faktor identifizieren. Hervorgehoben werden sollte in diesem Kontext, dass

das Szenario in vielen Fällen in Veranstaltungen des Grundstudiums erfolgreich durchgeführt wurde, in denen nicht in jedem Fall mit hoher studentischer Motivation zu rechnen ist. Das Einsatzspektrum für das vorgestellte Szenario ist daher durch didaktische Rahmenbedingungen kaum eingeschränkt.

### 3.3.2 Technische Anforderungen

Da die Wiedergabe der Aufzeichnung auf herkömmlichen PCs stattfindet, stellt dieses Szenario im Gegensatz zu den in 3.2.2 dargestellten Kriterien keine besonderen Anforderungen an die Aufzeichnungstechnik. Da die Studierenden das Material jedoch intensiv im Selbststudium nutzen, steigen die Ansprüche an die dramaturgische Gestaltung der Aufzeichnung. Das Material sollte frei von irrelevanten kontextbezogenen Äußerungen wie Kommentaren über das Wetter oder Aufrufen zur Ruhe sein. Darüber hinaus sollten veraltete Passagen ohne größeren Aufwand durch neuere Aufnahmen ersetzt werden können, ohne dass dabei die gesamte Sitzung neu aufgezeichnet werden muss. In vielen Erfahrungsberichten wird daher die Relevanz der Möglichkeit zur Nachbearbeitung der Aufzeichnung durch die verwendete Aufzeichnungs- bzw. Produktionssoftware betont [Smeaton & Keogh 1999; Mertens, Ickerott, Witte & Vornberger 2005; Moses, Litzkow, Foertsch & Strikwerda 2002; Stringer 2003].

Die Anforderungen an die technische Infrastruktur zur Datenhaltung und Bereitstellung der Aufzeichnung sind mit den in 3.1.2 beschriebenen Angaben für den ergänzenden Einsatz von Vorlesungsaufzeichnungen identisch, umfassen also Web- und ggf. Streamingserver. Wie auch in dem in 3.1 beschriebenen Szenario ist von der ausschließlichen Bereitstellung von Offline-Medien auf Datenträgern abzusehen. An der Universität Osnabrück wurde mit der Bereitstellung der in [Mertens, Ickerott, Witte & Vornberger 2005] beschriebenen Materialen als Offline-Medium ohne die Möglichkeit zum Download die Erfahrung gemacht, dass zwar hoch motivierte Studierende deutlich höhere Lernerfolge erzielt hatten als in vorangegangenen Semestern und auch als in dem in 3.2 beschriebenen zeitfixen Szenario, dass bei der Masse der Studierenden jedoch signifikant schlechtere Lernerfolge zu verbuchen waren.

Ist der Einsatz der Aufzeichnungen während begleitender Übungen und Tutorien geplant, wie in [Foertsch, Moses, Strikwerda & Litzkow 2002] dargestellt, stellen sich zusätzlich die dort beschriebenen Anforderungen an die Ausstattung der verwendeten Rechnerarbeitsräume. Hierbei steht jedoch nicht in erster Linie die technische Ausstattung sondern vielmehr das Mobiliar im Vordergrund. So sollten beispielsweise Tische und die Anordnung der Stühle auf die gruppenweise Nutzung von Vorlesungsaufzeichnungen zugeschnitten sein. Technisch sollte die Nutzung eigener Laptops mit Netzzugang über WLAN oder physikalische Verbindungen ermöglicht werden.

### 3.3.3 Organisatorische Rahmenbedingungen

Eine zentrale Frage bei der Planung des Szenarios ist die Vergabe von Scheinen über die erfolgreiche Teilnahme an der Veranstaltung. Da der eigentliche Lehrstoff in der Aufzeichnung vermittelt wird, muss der Lehrende nicht zwingend anwesend sein. Dies ist schließlich gerade einer der besonderen Vorteile der Veranstaltungsform. Wird das Szenario wie beispielsweise in [Effelsberg 2003] beschrieben eingesetzt, um eine Lehrveranstaltung durchzuführen, während der Dozent ein Forschungssemester in Anspruch nimmt, werden durch seine temporäre Abwesenheit keine Probleme verursacht. Wird das Szenario jedoch zum Import einer Lehrveranstaltung von einer anderen Universität verwendet, empfiehlt es sich, im Vorfeld Prüfungsbefugnisse zu klären und ähnlich geartete administrative Hürden zu beseitigen.

Darüber hinaus sind wie bereits in 3.3.1 angedeutet Übungen und Tutorien zu planen und die jeweils benötigten Räumlichkeiten zu organisieren. Ggf. ist dabei auf die Reservierung von Rechnerarbeitsräumen zu achten, die für Gruppenarbeit optimiert sind (vgl. 3.3.2).
Abgesehen von Veranstaltungen, die von anderen Universitäten importiert werden, hält sich der Organisationsaufwand daher in einem Rahmen, der mit dem einer herkömmlichen Lehrveranstaltung vergleichbar ist. Da Übungen in thematisch parallelen Sitzungen zu unterschiedlichen Terminen stattfinden können, wird die zeitliche Planung der Veranstaltungen in vielen Fällen sogar einfacher, da Terminkonflikte mit anderen Veranstaltungen durch die Möglichkeit eines Ausweichtermins leichter gelöst werden können.

### 3.3.4 Nutzenpotentiale

Zusammen mit einer kontinuierlichen und intensiven Betreuung der Studierenden kann die Verwendung von Vorlesungsmitschnitten als Ersatz der Vorlesung zu einer effektiven Gestaltung des Lernprozesses genutzt werden [Foertsch, Moses, Strikwerda & Litzkow 2002; Smeaton & Keogh 1999]. Durch die flexible Form der Wissensvermittlung ist eine Anpassung an den individuellen Wissenstand jedes einzelnen Studierenden möglich, wodurch die Lerneffizienz im Vergleich zu herkömmlichen Lehr-/Lernszenarien sogar gesteigert werden kann (siehe dazu 3.6.1). Hierbei ist jedoch zu berücksichtigen, dass das Szenario bisher immer nur als Einzelfall neben herkömmlichen Lehrveranstaltungen erprobt wurde. Ob die gesteigerte Lerneffizienz auch für den Fall des gleichzeitigen Einsatzes der Veranstaltungsform in mehreren Lehrveranstaltungen eines Studiensemesters gegeben ist, kann daher nur gemutmaßt werden.
Neben didaktischen Gründen kann auch eine Reihe von organisatorischen Faktoren für den Einsatz des Szenarios sprechen. So können beispielsweise terminliche Kollisionen vermieden werden oder eine Veranstaltung kann durchgeführt werden, obwohl der Dozent sich in einem Forschungssemester befindet [Effelsberg 2003]. Darüber hinaus ist das Szenario ähnlich wie auch das in 3.2 beschriebene Szenario gut zum Austausch von Veranstaltungen zwischen Universitäten geeignet.
Um das Verhältnis von Nutzen und Produktionsaufwand zu rechtfertigen, bieten sich vornehmlich Grundstudiumsveranstaltungen für das Szenario an, in denen relativ statisches Grundwissen vermittelt wird. Dadurch, dass die behandelten Themen in derartigen Veranstaltungen über mehrere Semester hinweg aktuell bleiben, kann eine Aufzeichnung mehrmals wieder verwendet werden. Dieser Punkt gewinnt im besprochenen Szenario insbesondere im Hinblick auf die in 3.3.2 erwähnte Notwendigkeit der Nachbearbeitung der Aufzeichnung an Relevanz.

### 3.3.5 Akzeptanz seitens der Studierenden

In Erfahrungsberichten wird die Akzeptanz des Szenarios seitens der Studierenden durchweg als sehr gut beschrieben. Gründe hierfür sind einerseits die intensivere Betreuung der Studierenden, wie in der von [Foertsch, Moses, Strikwerda & Litzkow 2002] beschriebenen Variation und andererseits Rahmenbedingungen, in denen den Studierenden der persönliche Nutzen klar wird, den sie selbst durch das Szenario erfahren. Derartiger zusätzlicher Nutzen ist beispielsweise mit der Durchführung einer Lehrveranstaltung trotz Forschungssemester des Dozenten [Effelsberg 2003] oder auch dann, wenn die Lehrveranstaltung in traditioneller Form aufgrund personeller Einschränkungen entfallen würde, gegeben.
Beim Vergleich der in [Foertsch, Moses, Strikwerda & Litzkow 2002] und [Stringer 2003] gemachten Erfahrungen wird klar, dass die Einbindung der Studierenden in Präsenzphasen deutlich zur Akzeptanz des Veranstaltungstyps beiträgt. Auch die Möglichkeit, in Gruppen zu arbeiten sowie die Einbindung in den curricularen Rhythmus sind der Akzeptanz des Szenarios durch die Studierenden förderlich. Zu betonen ist allerdings, dass die Akzeptanz

des Szenarios nicht zwingend positiv mit dem erzielten Lernerfolg korreliert. So hat der vergleichende Einsatz des hier beschriebenen und des in 3.2 dargestellten Szenarios an der Universität Osnabrück wie bereits in 3.2.1 eingehender dargestellt eine höhere Akzeptanz der zeitvariablen Präsentation des Materials gezeigt, führte jedoch zu einer signifikant niedrigeren Erfolgsquote der Studierenden.[13] In der Regel ist das Szenario jedoch neben einer hohen Akzeptanz durch die Studierenden mit deutlich besserem Lernerfolg der Studierenden verbunden [Foertsch, Moses, Strikwerda & Litzkow 2002; Smeaton & Keogh 1999].

### 3.3.6 Wiedergabemöglichkeiten

Um das Potential des zeitvariablen Szenarios voll nutzen zu können, sollte das eingesetzte Wiedergabemedium Möglichkeiten zur feingranularen Navigation bieten. So können beispielsweise bereits bekannte Themen übersprungen und schwierige Passagen beliebig oft wiederholt werden (vgl. dazu 3.6.1). Die Relevanz hochwertiger Navigationsmöglichkeiten im Rahmen dieses Szenarios zeigt sich auch in einer Reihe von Berichten, in denen eine in Bezug auf die Lerneffizienz oder Gesamtleistung der Studierenden erfolgreiche Durchführung des Szenarios dargestellt wird [Foertsch, Moses, Strikwerda & Litzkow 2002; Hurson & Kavi 1999; Smeaton & Keogh 1999]. In allen Fällen wurden Wiedergabeinterfaces mit erweiterten Navigationsmöglichkeiten verwendet.
In diesem Szenario ist außerdem die Möglichkeit, einzelne Abschnitte der Aufzeichnung direkt referenzieren zu können, besonders wichtig. Für einzelne Studierende ergibt sich daraus der Vorteil, eigene Linksammlungen zu erstellen, in denen sie individuell relevante Ausschnitte einzelner Aufzeichnungen verwalten können [Miyahara, Kaiya & Kaijiri 2002]. Weitere Nutzenpotentiale ergeben sich dann, wenn derartige Referenzen wie in [Mu 2005] beschrieben in Online-Kommunikationsmittel wie Foren oder Chats integriert sind. Die Referenzen können dann zusätzlich dazu eingesetzt werden, Fragen bezüglich bestimmter Abschnitte der Aufzeichnung zwischen Lehrenden und Studierenden aber auch Studierenden untereinander zu erörtern. Fragen können dabei gezielt zu einzelnen Punkten der Vorlesung gestellt werden. Im Gegensatz zu Fragen in einer Live Veranstaltung ist hierbei allerdings eine geringere Hemmschwelle seitens der Studierenden zu erwarten.
Darüber hinaus sind selbstverständlich auch in diesem Szenario die bereits in 3.1.6 angesprochenen generellen Qualitätsmerkmale zu berücksichtigen.

## 3.4 Vorlesungen on demand

In diesem Szenario verwenden einzelne Studierende verschiedene Vorlesungsaufzeichnungen unabhängig voneinander, um sich spezielles Wissen anzueignen oder um vor längerer Zeit erworbenes Detailwissen aufzufrischen. Das Einsatzspektrum reicht dabei von Prüfungsvorbereitungen über die gezielte Suche von Problemlösungsansätzen in Projekten oder im Beruf bis hin zur Erarbeitung von fehlendem Grundlagenwissen bei einem Hochschulwechsel. Im Gegensatz zu den anderen in diesem Kapitel beschriebenen Szenarien ist dieses Szenario nicht an eine bestimmte Veranstaltung gekoppelt, sondern besteht wie in Abb. 3-4 schematisch dargestellt aus der synchronen Bereitstellung von Aufzeichnungen mehrerer Veranstaltungen. Daraus er ergeben sich eine Reihe geänderter Rahmenbedingungen. So ist beispielsweise einer Aufzeichnung kein betreuendes Lehrpersonal zugeteilt und das vorhandene Material kann viele unterschiedliche Themenbereiche umspannen. Im folgenden Abschnitt werden daraus resultierende Anforderungen und mögliche Lösungsansätze diskutiert.

[13] Eine wahrscheinliche Erklärung für das schlechte Abschneiden des zeitvariablen Szenarios ist, wie bereits in Fußnote 12 auf Seite 40 erläutert, das Fehlen eines Online-Zuganges zu den bereitgestellten Materialien.

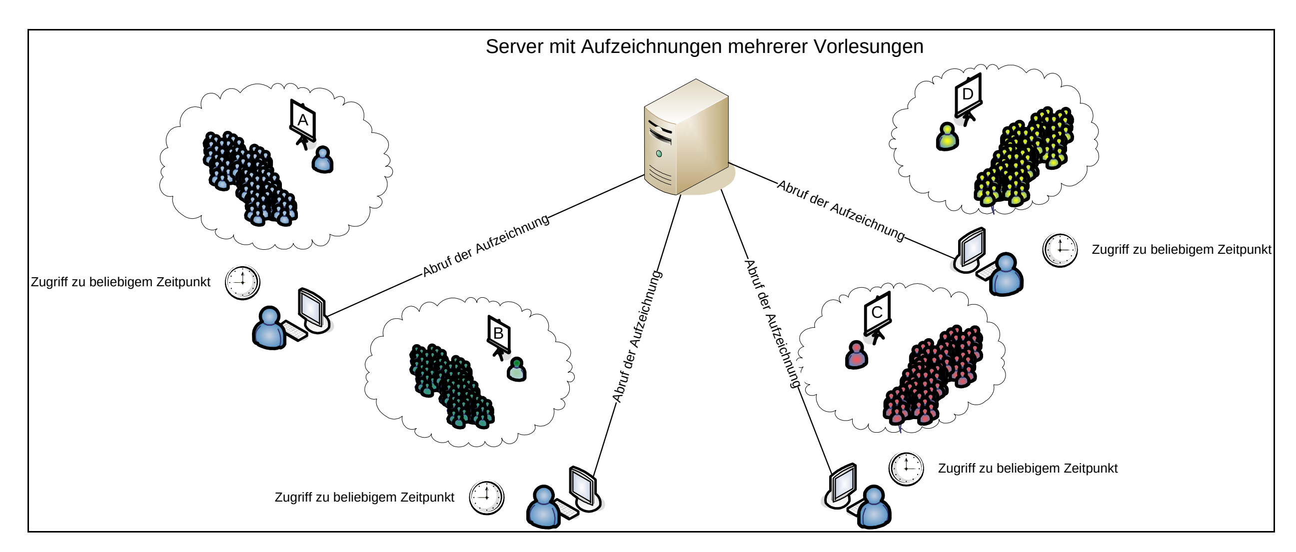



**Abb. 3-4: schematische Darstellung des Angebots von mehreren Online-Vorlesungen on demand**

### 3.4.1 Mediendidaktische Konzeption

Eine aktive didaktische Betreuung lässt sich in diesem Ansatz nur sehr schwer realisieren. Die Lehrkräfte und das Betreuungspersonal, das für eine Veranstaltung ursprünglich verantwortlich war, kann in der Regel aufgrund anderweitiger Lehrverpflichtungen hier nicht mehr herangezogen werden. Erschwerend hinzu kommt die Tatsache, dass Zugriffe auf einzelne Aufzeichnungen zu weit auseinander liegenden Zeitpunkten geschehen und dass die Aktionen und Lernziele einzelner Nutzer zeitlich nicht synchron sind. Die Möglichkeit zur Online-Kommunikation in Foren oder per Mail dürfte hier daher nur wenig hilfreich sein. Stellt ein Nutzer eine Frage im Forum, ist die Wahrscheinlichkeit, dass sich zeitnah ein anderer Nutzer mit derselben Thematik beschäftigt vergleichsweise gering. Aus diesen Gründen wäre die Erstellung von Materialien zur Selbstlernkontrolle im Rahmen einer didaktisch sinnvollen Gestaltung dieses Szenarios wichtiger als bei den voran genannten Szenarien.
In der Regel fällt das *on demand* Szenario jedoch als Nebenprodukt des Einsatzes von Vorlesungsaufzeichnungen in einem der voran beschriebenen Szenarien ab. In der Praxis sind Vorlesungsaufzeichnungen daher in den meisten Fällen nicht mit entsprechenden Zusatzmaterialien angereichert. Vorlesungsaufzeichnungen sind deshalb meist auch nicht als multimediale Selbstlernkurse einsetzbar [Effelsberg 2003] In der Zielsetzung des *on demand* Szenarios sind Sammlungen von Aufzeichnungen verschiedener Vorlesungen allerdings ohnehin eher als Informationssammlungen und nicht als aufbereitete Lernangebote zu verstehen.

### 3.4.2 Technische Anforderungen

Da die Durchführung der Aufzeichnung in diesem Szenario entfällt, beschränken sich die technischen Anforderungen auf die Bereitstellung der aufgezeichneten Materialien im Netz. Bei größeren Datensammlungen sind die Kapazitäten der verwendeten Videoserver zu berücksichtigen. Bei hohen Nutzerzahlen kann es ggf. durch Beschränkungen verwendeter Streamingserver zu Engpässen kommen.
Im Vergleich zu den bereits genannten Szenarien ist das Szenario jedoch mit einem minimalen technischen Aufwand verbunden.

### 3.4.3 Organisatorische Rahmenbedingungen

Prinzipiell müssen die organisatorischen Rahmenbedingungen des Szenarios hier nicht eingehender betrachtet werden, da sich der Aufwand auf die einmalige Bereitstellung des Materials auf einem Webserver beschränkt. Lediglich bei der Migration auf neue Server kann kurzzeitig Arbeitsaufwand entstehen.

Wird, wie in 3.4.1 angedeutet eine umfassende didaktische Betreuung der Studierenden angestrebt, ist hingegen eine Reihe von umfassenden organisatorischen Rahmenbedingungen zu schaffen. Hier stellt sich vor allem die Frage, wie der entstehende Betreuungsaufwand abgerechnet werden kann und wie er ggf. mit Lehrverpflichtungen verrechnet werden kann. Entsprechende Vergütungssysteme sind derzeit nicht angedacht. Weitere Probleme können sich an dieser Stelle dadurch ergeben, dass teilweise nur schwer vorhersehbar ist, welche Veranstaltung in welchem Zeitraum betreut werden muss und welchen Intensitätsgrad diese Betreuung erfordert.

### 3.4.4 Nutzenpotentiale

Die Nutzenpotentiale des Szenarios sind vielfältig. Das Szenario findet beispielsweise dann Anwendung, wenn sich Studierende in einem Semester, in dem die aufgezeichnete Veranstaltung nicht stattfindet, auf Prüfungen oder Nachklausuren vorbereiten. Nach einem Hochschulwechsel können Vorlesungsaufzeichnungen in der beschriebenen Form von Studierenden genutzt werden, um fehlendes Grundlagenwissen zu erarbeiten oder um unterschiedliche Schwerpunktesetzungen auszugleichen. In universitätsübergreifenden Projekten, wie in [Hasselbring, Appelrath, Sauer, Vornberger 2002] beschrieben, können so schnell curricular bedingte Wissenslücken in spezifischen Teilbereichen aufgefüllt werden. Ehemalige Studierende, die sich im Beruf befinden können gezielt nach Lösungsansätzen suchen, die sie im Studium vermittelt bekommen haben, an die sie sich aber nur noch vage erinnern können. Für diese Zielgruppe ist auch die Möglichkeit interessant, aktualisierte Teile einer besuchten Veranstaltung aufzurufen oder fachspezifisches Detailwissen aufzufrischen.

### 3.4.5 Akzeptanz seitens der Studierenden

In diesem Szenario steht die Verwendung der Aufzeichnung nicht mit einer betreuten Veranstaltung in Verbindung und hat in einigen Fällen noch nicht einmal einen Bezug zum aktuellen Curriculum. Im universitären Kontext ist die Bereitstellung von Vorlesungen *on demand* daher nur schwer zu evaluieren. Auch aufgrund der geringen Wartungskosten der einmal erstellten Mitschnitte kann eine aufwändige Evaluation dieses Szenarios nur schwer motiviert werden.

In Anbetracht der Nutzungsbedingungen der erstellten Materialien (vgl. 3.4.4) ist jedoch von einer hohen Eigenmotivation der Anwender auszugehen. Bei Akzeptanzuntersuchungen eines vergleichbaren Szenarios in der Wirtschaft hat sich gezeigt, dass die Bereitstellung von Aufzeichnungen *on demand* in der Regel sehr gut angenommen wird [He, Grudin & Gupta 2000; Zhang, Rui, Crawford, & He 2005].

### 3.4.6 Wiedergabemöglichkeiten

Aufgrund der Zielsetzung der Anwender, oft nur Details aus einer Aufzeichnung zu verwenden, ist die Möglichkeit zur gezielten Navigation innerhalb der Aufzeichnung essentiell [He, Grudin & Gupta 2000]. Auch die Möglichkeit zur externen Suche über mehrere Aufzeichnungen wie beispielsweise in [Ziewer 2005] oder AOF [Hürst 2002] vorgestellt kann entscheidend zur Effektivität der Nutzung einer Sammlung von Aufzeichnungen in diesem Szenario beitragen.

Soll das Material von berufstätigen Absolventen, dazu genutzt werden, aktuelle Änderungen im Lehrstoff einer vor längerer Zeit besuchten Veranstaltung nach zu verfolgen, sind bereits bei der Aktualisierung des Materials entsprechende Markierungen zu setzen. Systeme, die einen derartigen Workflow unterstützen, sind derzeit nicht geplant, stellen allerdings eine Perspektive für zukünftige Generationen von Vorlesungsaufzeichnungssystemen dar. In Anbetracht des Forschungsstandes auf dem Gebiet der Navigation in Vorlesungsaufzeichnungen, müssen jedoch zunächst die Grundsteine zur effektiven Navigation erarbeitet werden.

## 3.5 Erstellung von Vorlesungsaufzeichnungen

Der Begriff *Vorlesungsaufzeichnung* wird in der Regel mit einem Vorgehen in Verbindung gebracht, bei dem eine herkömmliche Vorlesung im Hörsaal aufgezeichnet wird. Darüber hinaus haben sich in den letzen Jahren jedoch auch alternative Formen der Erstellung von Vorlesungsaufzeichnungen entwickelt. Im weiteren Verlauf dieses Abschnitts werden kurz aktuelle Entwicklungen in diesem Bereich diskutiert, die im Hinblick auf die zuvor beschriebenen Einsatzszenarien relevant sind.

### 3.5.1 Integrierte Konzepte

Der Gedanke, dass sich einzelne Vorlesungsstunden gezielt bestimmten Abschnitten in einem Lehrbuch zuordnen lassen, ist in der universitären Lehre alltäglich. Vereinzelt finden sich jedoch mittlerweile auch in Lehrbüchern Verweise auf einzelne Online-Lernobjekte und Lehrbücher werden an aufgezeichnetes Material angepasst [Grossmann & Küchlin 2005]. Technisch lassen sich Querverweise von der Aufzeichnung ins Lehrbuch und von externen Medien in die Aufzeichnung realisieren [Mertens, Ickerott, Witte & Vornberger 2005].
Eine Aufzeichnung muss daher nicht als losgelöstes Produkt im Raum stehen, sondern kann zusammen mit anderen Lehrmaterialien wie (Online-) Übungsaufgaben, einem Lehrbuch oder Vorlesungsskript, einer Tonaufzeichnung für mobile Endgeräte usw. entwickelt werden. Die Inhalte der einzelnen Medien können dabei über die Grenzen des jeweiligen Mediums hinweg aufeinander verweisen und Teil eines einheitlichen Konzeptes sein.
Derartige Produktionsformen bedeuten für die Aufzeichnung eine nennenswerte didaktische Qualitätssteigerung, da die Nutzung der Aufzeichnung so beispielsweise auch mit ortsunabhängigen Lernformen wie dem Lesen eines Skriptes oder der Verwendung eines MP3-Players oder iPods verknüpft werden kann.[14] Im Gegenzug kann im Skript, Buch oder in der Tonaufzeichnung auf Teile der Videoaufzeichnung verwiesen werden, wenn deren Dynamik oder Visualisierung im jeweiligen Medium nicht hinreichend wiedergegeben werden kann.

### 3.5.2 Gestaltung der Aufnahme

Eine Vorlesungsaufzeichnung kann nicht nur durch die Aufnahme einer Live-Veranstaltung im Hörsaal, sondern auch durch eine Studioaufnahme [Krüger, Heinrich & Jobmann 2004], eine Aufnahme in einem Hörsaal, in dem keine Studierenden anwesend sind[15] oder an einem beliebigen Ort wie beispielsweise dem Arbeitszimmer des Dozenten produziert werden.
Für die Aufnahme im Studio sprechen Argumente wie Tonqualität, Beleuchtung, die Position der Kamera und die Möglichkeit, bestimmte Einstellungen beliebig oft aufzuzeichnen, um eine qualitativ hochwertige Aufnahme erstellen zu können [Effelsberg 2003]. Mittlerweile existieren jedoch an vielen Universitäten speziell ausgestattete Hörsäle, bei

[14] Ein Überblick zu Einsatzszenarien für Podcasts in der universitären Lehre wird in [Ketterl, Schmidt, Mertens & Morisse 2006] gegeben.
[15] Diese Bedingung kann auch bei einer Live-Veranstaltung auftreten. In diesem Fall sollten allerdings Begleitfaktoren wie beispielsweise die Terminwahl der Veranstaltung überprüft werden.

deren Einrichtung Faktoren wie Tontechnik und Ausleuchtung berücksichtigt wurden und die in der Regel mit fest installierten Kameras ausgestattet sind, so dass sich der Aufwand einer Studioaufzeichnung kaum lohnt.

Eine ähnlich hohe Flexibilität bei der zeitlichen Gestaltung und beim Ablauf der Aufzeichnung bietet die Aufzeichnung einer nachgestellten Veranstaltung in einem leeren Hörsaal oder im Arbeitszimmer. Auch hier können mehrere Sitzungen direkt hintereinander aufgezeichnet und beliebige Passagen wiederholt aufgezeichnet werden, da auch in dieser Gestaltungsform keine Studierenden einbezogen werden. Der Vorteil, dass sich diese Aufzeichnung ohne Studierende durchführen lässt, kann sich jedoch auch sehr schnell als Nachteil erweisen. In einer regulären Vorlesung kann ein Dozent an den Gesichtern der Studierenden ablesen, ob sie gerade gespannt zuhören, vom Stoff gelangweilt sind oder dem Tempo der Vorlesung nicht folgen können [Chen 2001]. Diese Faktoren sind zwar für die Aufzeichnung nicht von direkter Bedeutung und können darüber hinaus auch, wie im folgenden Abschnitt diskutiert, durch entsprechende Navigationsmöglichkeiten in der Aufzeichnung mehr als ausgeglichen werden, beeinflussen allerdings in der Regel die Dynamik des Lehrvortrags. So werden derartige Aufzeichnungen von Studierenden oft als weniger lebendig empfunden [Effelsberg 2003], wodurch in vielen Fällen auch ein wichtiger Bestandteil des Vortrags verloren geht. Von geschulten Sprechern kann dieser Nachteil sicherlich ausgeglichen werden, eine Vorlesungsaufzeichnung ist jedoch kein Lehrfilm und ein Hochschuldozent kein ausgebildeter Fernsehsprecher.

Im Vergleich mit den anderen Gestaltungsformen ist die Aufzeichnung einer Live-Veranstaltung am ehesten in den universitären Alltagsablauf integrierbar. Dozenten sind in der Regel routiniert genug, um auf die Möglichkeit, einzelne Passagen mehrmals aufzuzeichnen, verzichten zu können. Die durch die Aufzeichnung entstehenden Mehrkosten können minimal gehalten werden (vgl. 1.1.1). Rechtliche Probleme könnten sich dadurch ergeben, dass einzelne Studierende ihr Recht am eigenen Bild geltend machen. Dies könnte insbesondere dann kritisch werden, wenn es sich bei der aufzuzeichnenden Veranstaltung um eine Pflichtveranstaltung handelt. Derartige Fälle sind jedoch bisher an der Universität Osnabrück nicht aufgetreten. Auch in der Literatur sind hierzu keine Erfahrungen dokumentiert. Ein sehr häufiges Problem stellen die Zeiten für den Auf- und Abbau benötigter Technik wie der benötigten Kamera und weitere Hardware und eine eventuelle Anpassung der Tontechnik des Hörsaals dar. Besonders in Hinblick auf die oft vollen Belegungspläne gut ausgestatteter Multimediahörsäle und der Umstand, dass zwischen zwei Vorlesungen in einigen Fällen aufgrund von Terminanpassungen oder Überlänge weniger als 10 Minuten liegen, ergibt sich hier beim Einsatz mobiler Aufzeichnungstechnik oft ein Problem. Wenn möglich, sollte die Aufzeichnungstechnik daher in die feste Ausstattung des Hörsaals integriert werden. Darüber hinaus sollte die Aufzeichnungssoftware wie die im Rahmen dieser Arbeit entwickelte Software dazu geeignet sein, an eine derartige Infrastruktur angepasst zu werden.

Insgesamt sollte bei jeder der genannten Aufzeichnungsvarianten im Vorfeld ein Probelauf der Aufzeichnung unter Einbeziehung sämtlicher technischer Komponenten gefahren werden. Aufgrund der Tatsache, dass die Darstellung der Gestik von Vortragenden in einer Reihe von empirischen Studien durchweg positive Auswirkungen auf Lernverhalten und –Ergebnisse von Studierenden gezeigt hat [Kelly & Goldsmith 2004], sollte bei Videoaufzeichnungen auf eine zur Aufzeichnung von Gestik und Körpersprache geeignete Kameraperspektive geachtet werden. Wie sich in einer Untersuchung im Zusammenhang mit dem ePresence System (vgl. 5.4) gezeigt hat, ist die Erkennbarkeit derartiger Merkmale darüber hinaus auch zur Einordnung des Gesagten beispielsweise bei Unsicherheit des Vortragenden hilfreich [Dufour, Toms, Lewis & Baecker 2005].

Abschließend betrachtet ist der Aufzeichnung einer Live-Veranstaltung in den meisten Fällen der Vorzug gegenüber den anderen Varianten zu geben. Bei dieser Variante entsteht die Aufzeichnung quasi als Nebenprodukt einer regulär gehaltenen Lehrveranstaltung, ist weitaus authentischer als bei den anderen Aufzeichnungsformen und vermittelt die Dynamik der realen Veranstaltung.

## 3.6 Lerneffizienzsteigerungspotentiale durch Vorlesungsaufzeichnungen

Verschiedene Untersuchungen und Erfahrungen haben gezeigt, dass Vorlesungsaufzeichnungen an verschiedenen Punkten Potential zur Steigerung der Lerneffizienz von Studierenden aufweisen. Im Folgenden werden diese Punkte einzeln dargestellt, um eine Grundlage für die Diskussion in 3.7 zu liefern, in wie weit die jeweiligen Potentiale in den einzelnen Szenarien genutzt werden können und in wie fern sie bei der Gestaltung von Interfaces für die Wiedergabe von Vorlesungsaufzeichnungen berücksichtigt werden müssen.

### 3.6.1 Lerntempo und Stoffauswahl

Der Gedanke, dass Lehre dann am effektivsten ist, wenn sie auf das Lerntempo und den Kenntnisstand der Studierenden zugeschnitten ist, hat sich seit langer Zeit bewährt [Chen 2001]. In einer regulären Präsenzvorlesung sind jedoch in der Regel viele Studierende oft unterschiedlicher Fachrichtungen anwesend. Dies trifft gerade für gut besuchte Veranstaltungen im Grundstudium zu, in denen Grundlagenwissen vermittelt wird, das in verschiedenen Wissenschaftszweigen benötigt wird. Hier sind Vorwissen und fachspezifische Begabung der Studierenden in der Regel sehr unterschiedlich.

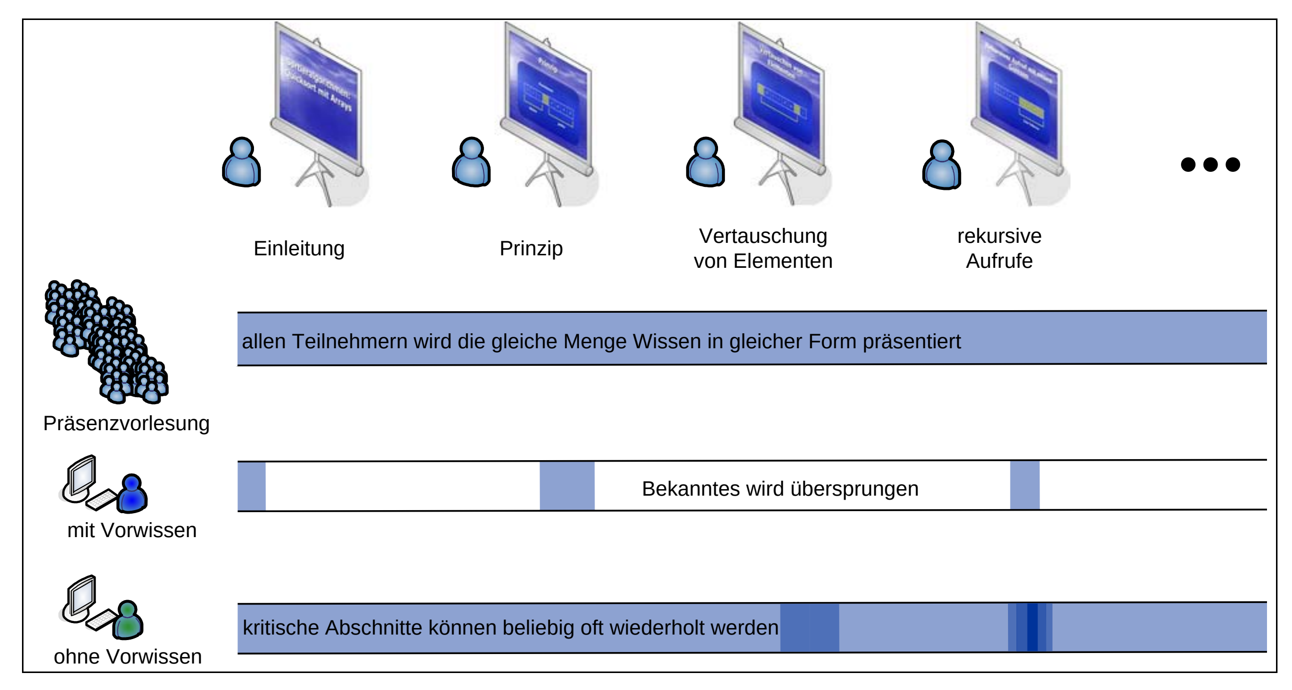



**Abb. 3-5: schematische Darstellung der Wissenspräsentation in Präsenzvorlesungen und bei der selbst gesteuerten Nutzung von Vorlesungsaufzeichnungen durch Studierende mit unterschiedlichem Vorwissen**

In einer Präsenzvorlesung wird das zu vermittelnde Wissen allen Studierenden in gleicher Form präsentiert. Vorlesungsaufzeichnungen bieten dem gegenüber den Vorteil, dass sich der Stoff der Vorlesung in kleine wiederholbare Einheiten unterteilen lässt und das Lerntempo individuell angepasst werden kann [Grossmann & Küchlin 2005]. Kritische oder

schwer verständliche Abschnitte können so beliebig oft wiederholt werden, bis ihr Inhalt verstanden wird. Darüber hinaus können sie auch im Zusammenhang mit anderen, zeitlich weiter entfernten Teilen der Aufzeichnung betrachtet und mit diesen direkt in Beziehung gesetzt werden. Ein weiterer Vorteil ergibt sich dadurch, dass bereits bekannte Inhalte übersprungen werden können. Diese beiden Faktoren wurden von [Hurson & Kavi 1999] in einer Auswertung verschiedener e-Learning-Szenarien als entscheidende Vorteile herausgearbeitet. Beobachtungen von [Smeaton & Keogh 1999] und [Foertsch, Moses, Strikwerda & Litzkow 2002] stützen diese Behauptung bezüglich des in 3.3 diskutierten Szenarios.
Abb. 3-5 verdeutlicht die Vorteile, kritische Abschnitte mehrmals zu wiederholen und die Präsentation von bereits bekanntem Wissen zu überspringen schematisch am Beispiel einer Einführung in den Sortieralgorithmus Quicksort mit Arrays. In der Vorlesung wird der gesamte Stoff allen Hörern einmal präsentiert. Bei Verwendung der Aufzeichnung können sich Studierende, die den Stoff nicht kennen, beispielsweise die Erklärung des prinzipiellen Ablaufes mehrmals anschauen. Studierende, die mit dem Stoff vertraut sind, können gezielt nach kritischen Abschnitten, wie beispielsweise der Auswahl der Grenzen für den rekursiven Abstieg suchen. Da Studierende in einigen Bereichen der Aufzeichnung über Vorkenntnisse verfügen können und in anderen Schwierigkeiten haben können, besteht in vielen Fällen auch die Möglichkeit, dass in einer Aufzeichnung von ein und demselben Studierenden kritische Abschnitte mehrmals angeschaut werden und andere übersprungen werden.

### 3.6.2 Tageszeit und Tagesform

Vorlesungsaufzeichnungen können nicht nur individuell an das jeweilige Lerntempo der Studierenden angepasst werden, sondern sind darüber hinaus auch ein asynchrones Lernmedium [Stern, Steinberg, Lee, Padhye & Kurose 1997]. Dieses Charakteristikum ist vor allem in Anbetracht der Tatsache relevant, dass die Lernleistung und Motivation von Studierenden oft auch abhängig von der Tageszeit, zu der eine Veranstaltung stattfindet und der jeweiligen Tagesform der Studierenden variiert. Oft sind Studierende zwar anwesend, können der Vorlesung aber nur schwer oder gar nicht folgen (vgl. dazu die in 3.2.1 dargestellten Erfahrungen). In der Vorlesung werden dann oft Übungsaufgaben abgeschrieben oder Studierende versuchen den Eindruck verständigen Lauschens zu erwecken. Gerade dann, wenn Veranstaltungen früh morgens oder zu anderen individuell ungünstigen Zeiten stattfinden, wirkt sich der Veranstaltungszeitpunkt negativ auf Lernleistung und Leistungsbereitschaft einzelner Studierender aus. Durch die Asynchronizität des Mediums Vorlesungsaufzeichnung kann dieser Nachteil ausgeglichen werden, da die Studierenden zeitflexibel auf das zu vermittelnde Wissen zugreifen können. Bei eventuell gehaltenen Übungen sollten in diesem Zusammenhang verschiedene Anfangszeiten angeboten werden.

### 3.6.3 Minimierung ablenkender Faktoren

In einer Vergleichsuntersuchung in [Glowalla 2004] wurden Präsenzveranstaltung, Vorlesungsaufzeichnung mit Video des Dozenten und Vorlesungsaufzeichnung mit einer Audiospur bezüglich der Akzeptanz seitens der Studierenden evaluiert. Eines der Ergebnisse der Untersuchung war die Tatsache, dass die Vorlesungsaufzeichnung mit Video von den Studierenden als am wenigsten ablenkend empfunden wurde. Eye-Tracking-Untersuchungen von Blickbewegungen bei der Verwendung von Folien mit Audio und Folien mit Video haben gezeigt, dass Folien mit Audio 60 Prozent der visuellen Aufmerksamkeit der Studierenden auf sich gezogen haben, wogegen Folien mit Video 90 Prozent der visuellen Aufmerksamkeit auf sich gezogen haben [Glowalla 2004]. Der Vorteil von Vorlesungsaufzeichnungen mit Video gegenüber Vorlesungsaufzeichnungen ohne Video und Präsenzveranstaltungen liegt also darin, dass die Vorlesungsaufzeichnung ablenkende Faktoren

minimiert und gleichzeitig durch die Videoaufnahme die Aufmerksamkeit auf das präsentierte Material fokussiert.
Selbst wenn Studierende bei Verwendung von Vorlesungsaufzeichnungen durch äußere Einflüsse abgelenkt werden, kommt der bereits in 3.6.1 erörterte Vorteil der Wiederholung von Abschnitten in Vorlesungsaufzeichnungen zur Geltung. Das Fehlen ablenkender Faktoren kann sich aber bei Vorlesungsaufzeichnungen auch nachteilig auswirken, wenn der Dozent den Lehrstoff nicht mit der gebotenen Dynamik vermittelt (vgl. dazu 3.5.2). Bei der Erstellung von Vorlesungsaufzeichnungen sollte daher mehr als in einer Präsenzvorlesung auf den Verlauf des Spannungsbogens geachtet werden, um diesen Vorteil der Vorlesungsaufzeichnung nicht zum Nachteil werden zu lassen.

## 3.7 Fazit

Vorlesungsaufzeichnungen ermöglichen es Studierenden, sich mit dem Lehrstoff einer Veranstaltung zeitflexibel, in selbstbestimmtem Lerntempo und mit geringst möglicher Ablenkung auseinanderzusetzen. Diese Vorteile sind in den dargestellten Szenarien in unterschiedlicher Ausprägung nutzbar. In diesem Kapitel wurden anhand von Einsatzerfahrungen und -berichten Empfehlungen erarbeitet, die zur erfolgreichen Umsetzung des jeweiligen Szenarios beitragen können.
Darüber hinaus war es Ziel der in diesem Kapitel dargestellten Betrachtung, Anforderungen an die Umsetzung von Interfaces zur Wiedergabe von Vorlesungsaufzeichnungen zu erarbeiten. Wie in 3.3.6 anhand eines Überblicks über verschiedene Einsatzberichte gezeigt, ist der in 3.6.1 erläuterte Nutzen, der sich durch die Möglichkeit, bekannte Abschnitte zu überspringen und kritische Punkte mehrmals zu wiederholen stark von den Navigationsmöglichkeiten des zur Wiedergabe verwendeten Interfaces abhängig. Die anderen durch Vorlesungsaufzeichnungen gebotenen Potentiale zur Steigerung der Lerneffizienz sind im Gegensatz dazu kaum durch die eingesetzte Software modulierbar. Auch die in 3.1.6 und 3.4.6 dargestellten Beobachtungen, sprechen für die Relevanz von ausgereiften Navigationskonzepten. In 3.2.6 wurde die Bedeutung einer hohen Darstellungsqualität der verwendeten Folien deutlich.
Im Rahmen der vorliegenden Arbeit werden beide Anforderungen umgesetzt. Zur Verbesserung der Darstellungsqualität von Folien bei der Wiedergabe wird das vektorbasierte Grafikformat SVG (Scalable Vector Graphics) verwendet. Um die Navigation in Vorlesungsaufzeichnungen zu erleichtern wird im weiteren Verlauf ein umfassendes Navigationskonzept für Vorlesungsaufzeichnungen entwickelt. Im folgenden Kapitel werden dazu zunächst bestehende Ansätze analysiert und Anforderungen an Navigationskonzepte für Vorlesungsaufzeichnungen herausgearbeitet.

# 4 Suche und Navigation in Vorlesungsaufzeichnungen

Beim Einsatz in der Lehre nehmen Vorlesungsaufzeichnungen oft die Rolle von Vorlesungsskripten oder Lehrbüchern ein [Zupancic & Horz 2002]. Wie im vorangehenden Kapitel festgestellt, werden Vorlesungsaufzeichnungen in den meisten Einsatzszenarien nicht als Film genutzt, der von Anfang bis Ende an einem Stück angeschaut wird. Vielmehr resultiert die Effizienz vieler Szenarien gerade daraus, dass Studierende sowohl relevante Ausschnitte gezielt heraussuchen und auch mehrmals betrachten können als auch irrelevante Passagen ignorieren können (vgl. 3.6.1). So können Studierende mit unterschiedlichem Vorwissen und unterschiedlichen Herangehensweisen ihre Vorlesung praktisch selbst zusammenstellen.
Die Tatsache, dass für verschiedene Studierende oft unterschiedliche und oft auch Detailfragen relevant sind, macht jedoch möglichst feingranularen Such- und Navigationsmechanismen unabdingbar [Hürst & Götz 2004]. Gerade in kursbegleitenden Einsatzszenarien, in denen die Studierenden die Aufzeichnung nur unterstützend nutzen, dient diese oft nur zur Wiederholung spezifischer Details, die der oder die Studierende nicht verstanden hat oder die zur Lösung einer konkreten Übungsaufgabe benötigt werden [Bell, Cockburn, McKenzie & Vargo 2001].
Aus dieser Notwendigkeit heraus sind eine Reihe von Navigationsansätzen entstanden, die in 4.2 betrachtet werden. 4.2.3 fasst Schwachstellen und Probleme dieser Ansätze zusammen. 4.3 stellt hypermediale Navigationskonzepte als Lösung zentraler Probleme aktueller Navigationsansätze vor und diskutiert deren Übertragbarkeit auf das Anwendungsgebiet Vorlesungsaufzeichnungen. 4.4 gibt einen detaillierten Überblick über hypermediale Navigationselemente für textbasierte Hypermedien. Darauf aufbauend werden in 4.5 Anforderungen an hypermediale Navigationskonzepte für Vorlesungsaufzeichnungen und daraus abgeleitet Implikationen zur Gestaltung geeigneter Benutzerinterfaces erarbeitet. Ein besonderes Augenmerk liegt dabei auf dem Unterschied zwischen traditionellen, textbasierten Hypermedien und Vorlesungsaufzeichnungen, die in erster Linie in die Kategorie zeitbasierte Medien einzuordnen sind. Einleitend wird in 4.1 kurz der Zusammenhang zwischen Navigation und Suche diskutiert, um die Verbindung zwischen den in Kapitel 3 erarbeiteten Anforderungen und den später betrachteten Navigationsansätzen zu verdeutlichen.
Einige der in diesem Abschnitt dargestellten Beobachtungen und Schlussfolgerungen wurden in kürzerer Form bereits in [Mertens 2003], [Mertens, Schneider, Müller & Vornberger 2004] und [Mertens, Brusilovsky, Ishchenko & Vornberger 2006] veröffentlicht.

## 4.1 Das Zusammenspiel von Suche und Navigation

Im Idealfall ist eine Vorlesungsaufzeichnung zeitlich in sehr kurze Passagen aufgeteilt. Diese Passagen sollten dabei jeweils Suchbegriffen zugeteilt sein, die den entsprechenden Ausschnitt möglichst genau beschreiben. Um die Suche effektiv zu gestalten, müssen diese Passagen nicht nur auffindbar, sondern auch direkt ansteuerbar sein [Brusilovsky 2000]. Derzeit existieren einige Systeme, die automatisch den Text der verwendeten Folien für Volltextsuche indizieren (siehe Kapitel 5). Der Nachteil der dort beschriebenen Systeme ist, dass die den betreffenden Folien zugeordneten Abschnitte von Anfang an abgespielt werden. Fällt der entscheidende Begriff erst später im Verlauf der betreffenden Folie, bedeutet dies entweder mehrere Minuten langes Warten oder Navigieren innerhalb der Folie. Eine Alternative stellt die manuelle Anreicherung der Aufzeichnung mit zeitlich hoch aufgelösten Metadaten [Einhorn, Nejdl & Olbrich 2003] dar. Das System ePresence bietet die Möglichkeit, in einem manuellen Arbeitsgang Suchbegriffe beliebigen Zeitpunkten zuzuordnen [Baecker, Wolf & Rankin 2004]. Derartige Nachbearbeitungsschritte sind allerdings mit einem nicht zu vernachlässigenden Arbeitsaufwand versehen. Dies widerspricht dem

*Authoring on the Fly* Gedanken zur kostengünstigen Erstellung von Vorlesungsaufzeichnungen [Bacher, Müller, Ottmann & Will 1997] und steht damit einem der Kernpunkte der Motivation zum Einsatz von Vorlesungsaufzeichnungen in der deutschen Hochschullandschaft entgegen (siehe 1.1.1). Im Regelfall ist sogar selbst die Eingabe grob aufgelöster Metadaten durch Dozenten nicht zu erwarten. Im Aufzeichnungssystem LectureLounge, das primär zur Aufzeichnung von Vorträgen im Gegensatz zu Vorlesungen konzipiert ist, beispielsweise werden Daten wie der Titel des Vortrags, Name des Vortragenden und ein Abstract über ein email-basiertes Vortrags-Benachrichtigungssystem gesammelt, um dieses Problem zu umgehen [Wolf, Putz, Stewart, Steinmetz, Hemmje & Neuhold 2004].
Aufgrund der mangelnden Indizierung sind zweckmäßige Suchfunktionalitäten in Vorlesungsaufzeichnungssystemen damit nur sehr schwer zu realisieren. Hinzu kommt die Tatsache, dass auch in textbasierten Medien reine Suchanfragen oft nicht zum Erfolg führen, da Benutzer falsche Suchanfragen stellen [Manber, Smith & Gopal 1997] oder die erwarteten Suchbegriffe nicht vorhanden sind. Bei der Suche in elektronischen Medien ist daher oft der Wechsel von konkreten Suchanfragen zu einem navigationsbasierten Suchverhalten zu beobachten, das einem Herantasten an gesuchte Ausschnitte ähnelt [Jul & Furnas 1997].
So berichten beispielsweise auch [Olston & Chi 2003], dass die Suche nach Informationen in multimedialen Umgebungen in einer Kombination aus Text-Suchanfragen und Browsing[16] besteht. Browsing ist vor allem für die Suche in nicht-textuellen Medien wie Audio- oder Videoaufzeichnungen geeignet, da diese Medien nur schwer für Suche zu indizieren sind [Cox, Haskell, LeCun, Shahraray & Rabiner 1998]. [Dunlop & Rijsbergen 1993] haben die Kombination von Text-Suche und Browsing als theoretischen Rahmen zur Implementation eines Suchalgorithmus für Bilder verwendet. Im Bereich Vorlesungsaufzeichnungen wird die Bedeutung von Browsing unter anderem durch Namen wie Cornell Lecture Browser [Mukhopadhyay & Smith 1999] und BMRC Lecture Browser [Rowe, Harley, Pletcher & Lawrence 2001] verdeutlicht. Der Autor Wolfgang Hürst widmet dem Thema „Suche durch Browsing“ in einer Dissertation zum Thema „Multimediale Informationssuche in Vortrags- und Vorlesungsaufzeichnungen“ [Hürst 2005] sogar mehr als 185 Seiten.
Browsing ist im Prinzip ein spezieller Navigationsstil [Sim, Clarke, Holt & Cox 1999]. Ein Vorlesungsaufzeichnungssystem, das Browsing unterstützt, muss also zunächst einmal Navigation unterstützen. Die im vorangehenden Absatz skizzierten Zusammenhänge zwischen Suche und Browsing existieren daher in gleicher Weise zwischen Suche und Navigation. Aufgrund dieser synergetischen Beziehung zwischen Such- und Navagationsfunktionalitäten werden im Folgenden Navigation und Suche nicht getrennt voneinander betrachtet, sondern als ein zusammenhängendes Thema behandelt.

## 4.2 Navigationsansätze für Vorlesungsaufzeichnungen

Zur Zeit existiert eine Reihe von Navigationsansätzen für Vorlesungsaufzeichnungssysteme. Im Folgenden werden diese Ansätze zunächst kurz vorgestellt. Daran anschließend werden verschiedene Grundströmungen identifiziert, die den derzeitigen Ansätzen zu Grunde liegen. Abschließend werden Schwachpunkte und Potentiale der herausgearbeiteten Grundströmungen diskutiert. In 4.3 wird hypermediale Navigation als ein Ausweg aus den Einschränkungen derzeitiger Ansätze diskutiert.

[16] Der Begriff *browsing* bezeichnet im Englischen das Überfliegen eines Dokumentes. Als Fachterminus im Bereich des Information Retrieval könnte er auch mit Herumstöbern o. ä. übersetzt werden. Prinzipiell ist der Begriff Browsing dort mit dem Begriff *hypermedia browsing* gleichzusetzen. Um Missverständnissen vorzubeugen, wird an dieser Stelle der englische Originalbegriff verwendet.

### 4.2.1 Aktuelle Ansätze zur Navigation in Vorlesungsaufzeichnungen

Der vorliegende Abschnitt gibt einen kurzen Überblick über aktuelle Entwicklungen zur Navigation in Vorlesungsaufzeichnungen. Die Ansätze werden hier nur kurz skizziert, eine ausführliche Darstellung der in diesem Zusammenhang wichtigsten Systeme ist in Kapitel 5 zu finden.

#### 4.2.1.1 Standardansätze

In den letzten Jahren sind mehrere Arbeiten angefertigt worden, die jeweils aus verschiedenen Perspektiven einen Überblick über den Stand der Forschung im Bereich Vorlesungsaufzeichnungen geben [Bijnens, Bijnens & Vanbuel 2004; Brusilovsky 2000; Lauer & Ottmann 2002; Knipping 2005; Sand 2003]. Übereinstimmend lässt sich zu den dort und zu den im Rahmen dieser Arbeit begutachteten Systemen zusammenfassen, dass fast alle derzeit verwendeten Vorlesungsaufzeichnungssysteme sowohl eine zeitbasierte Navigationsleiste als auch ein verlinktes Inhaltsverzeichnis bieten. Die zeitbasierte Navigationsleiste verfügt dabei in der Regel über einen Slider, mit dem beliebige Positionen innerhalb der Aufzeichnung angesteuert werden können. Bei dem verlinkten Inhaltsverzeichnis kann es sich klassisch um eine Liste der Folientitel oder um eine Thumbnail-Ansicht der verwendeten Folien handeln.
Das Fehlen eines verlinkten Inhaltsverzeichnisses bei einigen Systemen ist dadurch begründet, dass aufgrund des Aufnahmeverfahrens keine Strukturinformationen zur Verfügung stehen [Lauer & Ottmann 2002]. Eine erfreuliche Ausnahme bildet hier das TeleTeaching-Tool, in dem Strukturinformationen nach der Aufnahme durch heuristische Bildanalysetechniken erschlossen werden (vgl. 5.3).
Auch die Implementation einer Volltextsuche gestaltet sich bei vielen Implementationen recht schwer, da Folieninformationen im Web-Format vieler Systeme nicht symbolisch repräsentiert werden und damit auch die entsprechenden Textinformationen dort nicht mehr vorhanden sind [Lauer & Ottmann 2002]. Volltextsuche ist daher nur in vergleichsweise wenigen Systemen realisiert.

#### 4.2.1.2 Random Visible Scrolling

*Random visible scrolling* ist ein an der Albert-Ludwigs-Universität in Freiburg entwickelter Ansatz zur Navigation in Vorlesungsaufzeichnungen, der derzeit in den Systemen AOF und Lecturnity [Lauer & Ottmann 2002] und dem TeleTeachingTool (vgl. 5.3) eingesetzt wird. Bei diesem Ansatz wird Scrollen in der zeitbasierten Navigationsleiste des Systems ohne Zeitverzögerung im Folienfenster des Interfaces dargestellt [Bacher & Müller 1998].[17] Dadurch kann die gesamte Aufzeichnung beliebig beschleunigt sowohl vorwärts als auch rückwärts durchlaufen werden. Die Suche innerhalb der Aufzeichnung ähnelt dabei der Suche einer Szene in einem Videofilm durch Hin- und Herspulen des Videobandes.
Durch die in AOF und Lecturnity vorhandene Möglichkeit zur zeitsynchronen Wiedergabe von Mauszeigerbewegungen, handschriftlichen Annotationen auf dem Bildschirm und in begrenztem Umfang auch PowerPoint-Animationen lässt die Folienansicht der Aufzeichnung vergleichsweise genaue Rückschlüsse auf den Gegenstand der Vorlesung zum jeweiligen Zeitpunkt der Aufzeichnung zu.
Einen verwandten Ansatz stellt die in [Ponceleon & Dieberger 2001] beschriebe Navigationstechnik des *hierarchical brushing* dar, die zur Navigation in Videodaten entwickelt wurde. In dem zugrunde liegen Präsentationsformat wird das verwendete Videomaterial in Segmente zerlegt, die aus einer linearen Übersicht heraus ansteuerbar sind. Beim Überfahren einzelner Abschnitte der Übersicht mit dem Mauszeiger wird eine für das

[17] Bei Verwendung von Streaming Video ist das Scrollen technologiebedingt nur im Folienfenster, nicht im Videofenster möglich [Lauer & Ottmann 2002].

zugeordnete Segment charakteristische Momentaufnahme angezeigt. Zur feinkörnigen Navigation in größeren Videodateien wird der aktuelle Abschnitt in einer höheren Auflösung dargestellt, die in den Gesamtkontext eingebettet ist. Diese Vorgehensweise ist in Analogie zur Anwendung von *elastic panning* auf *random visible scrolling* zu betrachten, die im Rahmen der Entwicklung von AOF untersucht wurde (vgl. 5.2).
Prinzipiell sind sich beide Ansätze sehr ähnlich. Bei einer genügend hoch aufgelösten Segmentierung der Aufzeichnung sind beide Verfahren sogar nicht unterscheidbar. Im Gegensatz zu r*andom visible scrolling* wurde hierarchisches *brushing* jedoch bisher nicht zur Navigation in Vorlesungsaufzeichnungssystemen eingesetzt.

#### 4.2.1.3 Linking by interacting

Das *linking by interacting* Paradigma ist im Rahmen der Entwicklung von eClass/Classroom 2000 (vgl. 5.5) am Georgia Institute of Technology entstanden. Ein wesentlicher Bestandteil des Ansatzes ist die Nutzung handschriftlicher Annotationen zur direkten Ansteuerung der Filmsequenzen, in denen die Annotationen entstanden sind [Graca Pimentel, Abowd & Ishiguro 2000]. Da das System eine statische Übersicht über alle im Vortrag verwendeten Folien bietet und da die aufgezeichneten Annotationen auf den Übersichtfolien dargestellt werden, können Annotationen durch manuelles Durchsuchen der Folienübersicht gefunden werden. Mit einem Mausklick auf die entsprechende Annotation wird der Film ab der Position abgespielt, an der die Annotation aufgezeichnet wurde.
Dieser Ansatz kann selbstverständlich abstrahiert werden, so dass auch in PowerPoint animierte Objekte zu verlinkten Orientierungspunkten auf den Übersichtsfolien werden. In eClass/Classroom 2000 können PowerPoint-Folien jedoch nur als statische HTML-Dateien importiert werden [Brotherton 2001]. Alle auf den Folien enthaltenen Animationen gehen dadurch verloren und können daher nicht zur Erzeugung einer Linktabelle genutzt werden. Im Rahmen der hier vorliegenden Arbeit ist der Grundgedanke der Verlinkung von Benutzerinteraktionen auf PowerPoint-Animationen erweitert worden. Aufgrund der dadurch gewonnenen Verzahnung von Feinstruktur des Vortrags und symbolischer Repräsentation der Inhalte werden, wie im weiteren Verlauf der Arbeit beschrieben, eine Vielzahl weiterführender Navigationshilfen möglich.

#### 4.2.1.4 Weitere Ansätze

Neben *random visible scrolling* und der in 4.2.1.3 beschriebenen Implementation des *linking by interacting* Paradigmas sind im Rahmen der Entwicklung anderer Systeme zur Vorlesungsaufzeichnung noch eine Reihe weiterer Navigationsmechanismen entstanden. Diese sind teilweise in Antizipation der Anforderungen an die Systeme in bestimmten Lehr-/Lernszenarien [Schapira, De Vries & Pedregal-Martin 2001; Baecker 2003], aber auch aufgrund der Evaluation der Systeme im Einsatz entstanden [Moses, Litzkow, Foertsch & Strikwerda 2002].
Es handelt sich bei diesen Mechanismen in der Regel nur um kleinere, voneinander unabhängige Navigationsfunktionen. So verfügt beispielsweise eTEACH über eine Funktionalität zur Markierung bereits betrachteter Abschnitte der Aufzeichnung (vgl. 5.6.2). Im BMRC Lecture Browser wird die Länge aller Folienabspielzeiten in der Zeitleiste angezeigt. Das System ePresence bietet eine interaktive *timeline*, auf der Folieneinstiegspunkte markiert sind. Darüber hinaus bieten einige wenige Systeme die Möglichkeit, anhand der Zeitpunkte von Folienanimationen zu navigieren [Brusilovsky 2000].
Eine eingehendere Betrachtung der verschiedenen Ansätze und der zugehörigen Systeme erfolgt in Kapitel 5. Im Zusammenhang mit der Beschreibung hypermedialer Navigation in 4.4 wird deutlich, dass viele der beschriebenen Ansätze konzeptionell dichter zusammen-

hängen, als eine oberflächliche Betrachtung zunächst vermuten lässt (vgl. auch die Einleitung zu Kapitel 5).

### 4.2.2 Grundströmungen

Insgesamt lassen sich bei Betrachtung der oben kurz beschriebenen und der in Kapitel 5 eingehender dargestellten Navigationsansätze mehrere Grundströmungen erkennen, zu deren Einordnung die Kriterien Zeit, Struktur und Verwendung handschriftlicher Annotationen herangezogen werden können.
Im Folgenden werden zunächst kurz der Zusammenhang zwischen Zeit und Struktur sowie das Verhältnis zwischen zeit- und strukturbasierter Navigation beleuchtet. Daran anschließend werden die Eigenschaften von annotationszentrierten Navigationsansätzen und dokument- und folienstrukturbasierten Ansätze gegenübergestellt.
Der Abschnitt soll dabei nicht nur der Beschreibung der einzelnen Grundströmungen dienen, sondern auch Potentiale und Probleme aufzeigen.

#### 4.2.2.1 Zeit und Struktur

In multimedialen Dokumenten, in denen zusammen mit statischen Objekten wie Bildern und Folien auch zeitabhängige Medien wie Audio und Video verwendet werden, kann benutzerseitige Interaktion mit dem Dokument und damit auch Navigation sowohl zeit-, als auch strukturbasiert realisiert werden [Hardman, Schmitz, van Ossenbruggen, ten Kate & Rutledge 2000]. Da Zeit und Struktur in den meisten Systemen zur Vorlesungsaufzeichnung durch eine zeitlich harte Synchronisation[18] sehr eng an einander gekoppelt sind, sind dort auch zeit- und strukturbasierte Navigation kaum zu trennen. So ist beispielsweise mit dem eher zeitbasierten Ansatz des *random visible scrolling* in beschränktem Maße auch strukturbasierte Navigation möglich. Durch die verzögerungsfreie Aktualisierung der Folienansicht bei benutzerseitiger Manipulation des Zeitsliders kann *random visible scrolling* wie das Blättern in einem Buch dazu genutzt werden, strukturelle Elemente der verwendeten Folien durch manuelle Suche zu lokalisieren (vgl. dazu 4.3.4.1).[19]
Bei herkömmlichen strukturbasierten Navigationsansätzen wird die Verbindung zwischen Zeit und Struktur in der Regel erst nach Durchführung der Navigationsaktion wie dem Anklicken eines maussensitiven Folientitels durch den Sprung zum jeweils angewählten Abspielzeitpunkt offensichtlich. In vergleichsweise vielen Systemen wie beispielsweise dem BMRC Lecture Browser [Rowe, Harley, Pletcher & Lawrence 2001], dem Cornell Lecture Browser [Mukhopadhyay & Smith 1999] dem Enhanced Viewer Experience System [Mate, Velasquez & Sarmenta 2004], eTEACH (siehe 5.6), Just-In-Time Lectures [Dannenberg & Capell 1997], Lecturnity [Lauer & Ottmann 2002] und dem TeleTeachingTool (siehe 5.3) wird der Einsprungpunkt des an den Titel oder die Thumbnail-Darstellung der Folie gekoppelten Links textuell in Form eines Zeitindex dargestellt.
Der BMRC Lecture Browser, die Wiedergabekomponente des PRISM Toolkits [Cuthbert, Kubinec, Tanic, Ieong, Wei & Schlossberg 2005] und der Cornell Lecture Browser bieten darüber hinaus eine Zeitleiste, in der die Spieldauer der verwendeten Folien durch vertikale Striche markiert ist. Der Zeitabschnitt, der der aktuell gezeigten Folie zugeordnet ist, wird im BMRC Lecture Browser rot markiert. Diese Navigationsunterstützung ermöglicht daher eine Orientierung über die zeitliche Position und die Abspieldauer der aktuell gezeigten Folie. Prinzipiell würde diese Art der Visualisierung auch eine Orientierung über die zeitliche

[18] Der Begriff zeitlich harte Synchronisation wird hier nach der Definition von [43] verwendet. Dabei wird zwischen harter und weicher Synchronisation unterschieden. Bei hart synchronisierten Medien darf beim Abspielen zweier Medien kein merklicher Zeitversatz auftreten.
[19] In der Praxis ist dies nicht in jedem Fall möglich. So werden beispielsweise in AOF und Lecturnity einige Animationen falsch, gar nicht oder in der falschen Reihenfolge wiedergegeben (vgl. 5.2).

Einordnung aller Folien ermöglichen, dazu müsste der Nutzer die Zuordnung von Folie zu Zeitintervall jedoch durch Abzählen der Intervallmarkierung bewerkstelligen. Eine Lösung dieses Problems stellt die interaktive *timeline* des Systems ePresence (siehe 5.4) dar. Dort sind zeit- und strukturbasierte Navigation in einem Navigationselement vereint. Auf der *timeline* werden alle Zeitpunkte, zu denen eine neue Folie angezeigt wird als vertikale Linien auf der Zeitachse markiert. Bei Überfahren einer solchen Markierung mit dem Mauszeiger wird der Titel der jeweiligen Folie angezeigt. Durch Anklicken beliebiger Positionen in der *timeline* kann zu den Zeitpunkten in der Aufzeichnung gesprungen werden. Ein Nachteil dieses Ansatzes ist, dass er nur unidirektional zeitbasierte Navigation mit Strukturinformationen anreichert. So hat eine Evaluation von ePresence gezeigt, dass die *timeline* von Nutzern bei vielen Suchaufgaben zugunsten strukturbasierter Navigationselemente vernachlässigt wird, da strukturbasierte Navigationselemente eine Vielzahl zusätzlicher Informationen beinhalten [Dufour, Toms, Lewis & Baecker 2005].
Insgesamt fällt auf, dass nur recht wenige verschiedene Ansätze zur Visualisierung der Verbindung zwischen zeit- und strukturbasierter Navigation existieren. Die verhältnismäßig große Anzahl der Systeme, in denen versucht wird, eine Visualisierung dieser Verbindung zu implementieren, zeigt jedoch die Relevanz dieses Vorhabens. Darüber hinaus ist anzumerken, dass in bisherigen Ansätzen Zeit und Struktur nur unidirektional miteinander verbunden wird. Entweder wird zeitbasierte Navigation mit Strukturinformationen angereichert oder strukturbasierte Navigationselemente werden durch eine zeitliche Einordnung erweitert. Ein Ansatz, der eine integrierte, bidirektionale Verbindung von Zeit und Struktur herstellt, ist bisher in keinem in der Literatur bekannten System zur Wiedergabe von aufgezeichneten Vorlesungen implementiert.

#### 4.2.2.2 Strukturbasierte versus annotationszentrierte Ansätze

In einigen Systemen zur Vorlesungsaufzeichnung existiert neben der durch Reihenfolge und Aufbau der Folien gegebenen Strukturierung der Aufzeichnung eine weitere Möglichkeit zur Markierung von Kernpunkten innerhalb der Aufzeichnung. In Systemen wie beispielsweise AOF oder eClass können, wie bereits in 4.2.1.2 und 4.2.1.3 erwähnt, handschriftliche Annotationen, die während der Aufnahme aufgezeichnet werden, zusammen mit dem Folienmaterial gespeichert werden. So können Folien handschriftlich ergänzt oder relevante Folienbestandteile hervorgehoben werden.
Besonders in Systemen, in denen Animationen nicht oder nicht zuverlässig funktionieren, tendieren Lehrende dazu, relevante Folienbestandteile einzukringeln oder in ähnlicher Art und Weise zu markieren. Die Entwickler einiger Systeme wie E-Chalk [Knipping 2005] oder Lectern II [Joukov & Chiueh 2003] sind sogar so weit gegangen, Aufzeichnungssysteme für reine Tafelvorlesungen zu entwickeln. Bezeichnenderweise existieren jedoch auch in diesen Systemen Importroutinen zur Einbindung von PowerPoint-Präsentation.[20]
Je mehr die Aufzeichnung einem Tafelvortrag gleicht, desto schwieriger werden Navigation und Orientierung anhand von Aufbau und Struktur der Folien. In diesen Fällen zeigen die in 4.2.1.2 und 4.2.1.3 beschriebenen Ansätze in der in AOF/Lecturnity bzw. eClass implementierten Form als annotationszentrierte Ansätze ihre Stärken. Da die Implementationen stark auf die Navigation anhand von Mauszeigerinteraktion und handschriftlichen Annotationen ausgerichtet sind, kann hier prinzipiell auch ohne Rückgriff auf die Dokumentstruktur navigiert werden. In der Praxis sieht es jedoch in sehr vielen Fällen anders aus. Dort werden

[20] In Lectern II wird die Windows-Druckausgabe zum Import genutzt [Joukov & Chiueh 2003]. PowerPoint-Animationen werden daher nicht berücksichtigt (vgl. 8.1). Eine Testversion der Konvertierungssoftware *ppt2echalk* für das System E-Chalk ist derzeit nicht erhältlich. [Knipping 2005] schildert exportierte Folien, beschreibt diese jedoch als statische Bilder. Auch hier werden Animationen also nicht berücksichtigt.

Annotationen oder Mauszeigerposition oft nur zur Markierung von Folienelementen genutzt [Anderson, Hoyer, Wolfman & Anderson 2004], wie in Abb. 4-1 dargestellt.

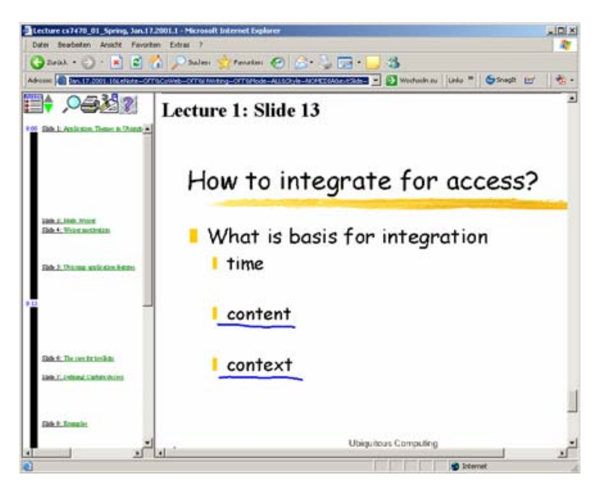



**Abb. 4-1: Manuelle Markierung struktureller Folienelemente am Beispiel eines eClass-Vortrags**[21]

Besonders in Systemen wie eClass, in denen Folienanimationen nicht wiedergegeben [Brotherton 2001], oder AOF und Lecturnity, in denen eine Reihe von Animationen nicht nachgebildet werden können (vgl. 5.2), versuchen Lehrende so die Aufmerksamkeit der Hörer zu lenken. Da eine genauere Betrachtung der Verwendung von PowerPoint, Whiteboards und Grafiktabletts aus didaktischer Sicht bereits in 2.2 stattgefunden hat, sei zum Vergleich der zugehörigen Lehrstile auf diesen Abschnitt verwiesen.
Die Nachteile der Nutzung von Annotationen zur Navigation werden vor allem im Vergleich mit strukturbasierten Ansätzen deutlich. So ist beispielsweise Volltextsuche in Annotation kaum zu realisieren [Knipping 2005]. Als großes Problem bei der Worterkennung erweist sich dabei die Tatsache, dass gerade für die Suche relevante Stichworte aufgrund ihres seltenen Vorkommens oft von der Erkennungssoftware verworfen werden [Theimer 2004].[22]
Auch in hybriden Ansätzen, in denen Folientext und Annotationen zusammen genutzt werden, ist Volltextsuche oft wenig erfolgreich. Da Folien beim Einsatz derartiger Systeme in vielen Fällen ganz bewusst mit Lücken werden und erst während des Vortrags durch Annotationen vervollständigt werden [Bacher & Ottmann 1995],[23] sind dort verwendete Suchbegriffe oft nicht im durchsuchbaren Folientext enthalten.
Bei der manuellen Suche in handschriftlichen Annotationen stellt i.d.R. die krakelige Schrift ein Hindernis dar. Besserung versprechen hier neu entwickelte Hardware und Eingabegräte [Ottmann & Lauer 2002] sowie verbesserte Aufnahme- und Wiedergabesoftware [Mohamed, Belenkaia & Ottmann 2004; Knipping 2005]. Auch bei Einsatz von Whiteboards und Grafiktabletts ist jedoch oft schlechte Schriftqualität zu beobachten. Hinzu kommt, dass handschriftliche Annotationen in vielen Fällen nicht als Einstieg in einen thematischen Unterpunkt sondern als abschließendes Fazit genutzt werden und so kein direktes Ansteuern eines zugeordneten Abschnittes der Aufzeichnung ermöglichen [Brotherton & Abowd 2004].

[21] Screenshot entnommen aus [9]
[22] Durch Nachbearbeitung der entsprechenden Wortlisten ist sicherlich in einigen Fällen eine Verbesserung der Erkennungsleistung zu erreichen. Auch eine manuelle Nachindizierung der Dokumente ist machbar. Beide Vorgehensweisen widersprechen allerdings dem Grundgedanken der nebenläufigen Content-Erzeugung durch Aufzeichnung einer Lehrveranstaltung.
[23] Entscheidende Folienbestandteile erst während des Vortrags auf der Folie erscheinen zu lassen, ist didaktisch durchaus sinnvoll [Weidenmann 2002]. Wenn dies durch Verwendung von PowerPoint-Animationen bewerkstelligt wird, ist eine Volltextsuche, wie im weiteren Verlauf der Arbeit gezeigt, ohne größere Probleme realisierbar.

Strukturbasierte Ansätze haben demgegenüber den Vorteil, dass eine sehr gute Schrift- und Darstellungsqualität erreicht werden kann, die auch die manuelle Suche innerhalb der Präsentation erleichtert.[24] Auch Volltextsuche ist hier weitaus leichter realisierbar. Darüber hinaus können strukturelle Elemente wie Bullet-Listen oder die Animationsreihenfolge von Folienbestandteilen genutzt werden (vgl. 6.2.2). Der Vorteil der Übersichtlichkeit wird beispielsweise in einer Studie über das Navigationsverhalten von Studierenden in eClass deutlich. In eClass kann sowohl anhand der Folienstruktur als auch anhand von Annotationen navigiert werden kann. In der Studie ist festgestellt worden, dass dort Folien doppelt so häufig genutzt werden wie Annotationen [Brotherton & Abowd 2004]. Dieses Resultat ist umso bemerkenswerter in Anbetracht der Tatsache, dass in eClass keine feinkörnige Navigation anhand der Folien möglich ist und dass in anderen Arbeiten und Studien zum Nutzerverhalten in Vorlesungsaufzeichnungen besonders die Relevanz von feinkörnigen Navigationsmöglichkeiten hervorgehoben wird [Brusilovsky 2000; Moses, Litzkow, Foertsch & Strikwerda 2002; Hürst & Götz 2004].
Auch der Erfolg beim Einsatz strukturbasierter Systeme wie MANIC und eTEACH in Lehr-/Lernszenarien, in denen eigenständige Suche und Navigation im aufgezeichneten Material entscheidend für den Lernerfolg der Studierenden ist [Schapira, De Vries & Pedregal-Martin 2001; Foertsch, Moses, Strikwerda & Litzkow 2002], ist ein Indiz für die Effektivität strukturbasierter Ansätze zur Navigation in Vorlesungsaufzeichnungen. Als nachteilig erweisen sich bei vielen strukturbasierten Systemen umständliche Produktionsprozesse (vgl. 5.1 und 5.6) und der Umstand, dass Synchronisationsdaten manuell eingefügt werden müssen (vgl. 5.4 und 5.6). Dank der Einführung einer COM-Schnittstelle für PowerPoint, ist es jedoch mittlerweile auch möglich, während eines Live-Vortrags Interaktionsdaten wie Folienwechsel zu protokollieren und zur automatischen Indizierung von Vorlesungsaufzeichnungen zu nutzen [Mühlhäuser & Trompler 2002]. Neben dem Import von PowerPoint-Präsentationen in spezielle Software-Tools wie AOF oder Lecturnity (vgl. 5.2), die Interaktionsdaten protokollieren können, und der manuellen Einarbeitung dieser Daten existiert damit ein dritter Weg zur Nutzung struktureller Informationen einer Präsentation. Nachteile wie Importfehler oder manuelle Nachbearbeitung entfallen dabei.
Die Vorteile strukturbasierter Ansätze sind daher mittlerweile nutzbar, ohne dass Einbußen beim Produktionsprozess entstehen.

### 4.2.3 Fazit

Die Analyse bisher verwendeter Ansätze zur Navigation in Vorlesungsaufzeichnungen zeigt eine Reihe interessanter Entwicklungen, weist jedoch auch auf entscheidende Entwicklungspotentiale hin.
Zunächst ist festzustellen, dass das Verhältnis zwischen Zeit und Struktur in bisherigen Ansätzen nur unzureichend visualisiert wird. Ein Problembewusstsein ist bei vielen Entwicklern vorhanden, es existieren jedoch nur zwei unterschiedliche Lösungsansätze, die das Problem zudem nur rudimentär lösen.
Ein weiteres Ergebnis der Betrachtung vorhandener Implementationen ist die Aufteilung in annotationszentrierte und strukturbasierte Ansätze. Da Annotationen und Struktur in der Regel sehr eng miteinander korrelieren, sind annotationszentrierte Ansätze oft auch zur strukturbasierten Suche nutzbar.
Insgesamt zeigt der im vorangehenden Text dargestellte Vergleich strukturbasierter und annotationszentrierter Strategien in vielen Einsatzszenarien einen deutlichen Vorteil struktur-

[24] In vielen Systemen, in denen strukturbasierte Navigation eingesetzt wird, wird dieser Vorteil durch schlechte Bildqualität und zu kleine Übersichtsfolien wieder zunichte gemacht [Hürst & Götz 2004]. Der Fehler liegt dort in der konkreten Implementation, nicht in den Ansätzen selbst.

basierter Ansätze. Dem gegenüber steht allerdings bislang kein umfassendes strukturbasiertes Such- und Navigationskonzept.
Diesen Ergebnissen folgend wird im nächsten Abschnitt das Potential hypermedialer Navigation zur Suche und Navigation in Vorlesungsaufzeichnungen diskutiert. Dazu wird zunächst der Zusammenhang zwischen Hypermedien und Dokumentstrukturierung näher beleuchtet. Daran anschließend wird der Unterschied zwischen statischen und zeitabhängigen Medien betrachtet, um im weiteren Verlauf eine konzeptuelle Anpassung bisheriger hypermedialer Navigationskonzepte, die für statische Textmedien entwickelt wurden, zu ermöglichen.

## 4.3 Eignung hypermedialer Navigation als Ansatz zur Navigation in Vorlesungsaufzeichnungen

Die vorangehenden Abschnitte dieses Kapitels haben illustriert, dass sowohl Möglichkeiten zur strukturbasierten Navigation als auch Funktionalitäten, die das Browsing in Vorlesungsaufzeichnungen unterstützen, Suche und Navigation in Vorlesungsaufzeichnungen entscheidend verbessern können. Ein Ansatz zur Suche in elektronischen Medien, der in den letzten Jahren sehr stark an Bedeutung zugenommen hat, ist die hypermediale Aufbereitung der enthaltenen Information. Hypermedien bieten eine verzeigerte, nicht-lineare Struktur, anhand derer sich Nutzer orientieren können. Durch die Nicht-Linearität des Mediums wird eine hohe Flexibilität erreicht, die Hypermedien zu einem effektiven Werkzeug für das Browsing in Informationsräumen werden lassen [Nielsen 1990a]. Im Forschungsfeld Vorlesungsaufzeichnungen wird in der neueren Überblicksliteratur der Ruf nach einer verlinkten Struktur der Aufzeichnung [Brusilovsky 2000] oder zumindest einem verlinkten, mit Thumbnails visualisierten Inhaltsverzeichnis [Lauer & Ottmann 2002] laut. Beide Forderungen stellen einen wichtigen Schritt in Richtung der Transformation von Vorlesungsaufzeichnungen in hypermediale Dokumente dar.
Um der Frage nachzugehen, in wie weit sich hypermediale Navigation als Navigationsansatz für Vorlesungsaufzeichnungen eignet, zeigt dieser Abschnitt zunächst, dass die Konzepte, die hypermedialer Navigation zugrunde liegen, beide der oben genannten Anforderungen, also Unterstützung von strukturbasierter Navigation und Browsing, erfüllen. Daran anschließend werden Probleme der Übertragbarkeit der Grundkonzepte hypermedialer Navigation auf das Anwendungsfeld Vorlesungsaufzeichnungen diskutiert.

### 4.3.1 Hypermediale Navigation als strukturbasierte Navigation

Die Nutzung der Struktur hypermedialer Dokumente zur Navigation und Orientierung ist ein effektiver Weg zur Vereinfachung dieser beiden Aufgaben [Hannemann, Thüring & Haake 1993]. Mittlerweile ist der Einsatz struktureller Elemente zur Navigation in Hypermedia sogar fester Bestandteil der Definition des Begriffes Hypermedia [Bieber 2000]. Darüber hinaus stellen Inhaltsverzeichnisse und strukturelle Überblicke weitere strukturbasierte Orientierungshilfen dar [Bernstein, Brown, Frisse, Glushko, Landow & Zellweger 1991]. Verlinkte Inhaltsverzeichnisse erlauben eine direkte Nutzung der so dargestellten Information. Durch eine farbliche Markierung der Links, die so genannte Footprint-Navigationshilfe (siehe 4.4.2.3), werden über die Links erreichbare Teile eines Dokumentes als gelesen oder ungelesen markiert. Dieses Vorgehen kann als eine zusätzliche Strukturierung des jeweiligen Dokumentes verstanden werden. Durch vom Benutzer selbst definierte Linklisten im Browser, so genannte Bookmarks (siehe 4.4.2.5), können individuell strukturierte Sichten auf ein Dokument erstellt werden.[25]

[25] Die Strukturierung der Bookmarks kann im Browser durch eine selbst definierte Ordnerstruktur verfeinert werden. Das im weiteren Verlauf der Arbeit vorgestellte Interface unterstützt keine solche Ordnerstruktur,

### 4.3.2 Browsing und Hypermedien

Wie bereits in 4.1 erörtert, spielt Browsing bei der Suche in Vorlesungsaufzeichnungen eine entscheidende Rolle. Der Begriff Browsing in seiner heutigen Verwendung stammt aus dem Bereich Hypertext. Der Oberbegriff für Software wie den Netscape Navigator oder den Internet Explorer lautet Browser, weil die Aufgabe dieser Programme darin besteht, Nutzer beim Browsen in Hypertext und statischen Hypermedien wie Bildsammlungen zu unterstützen. Gerade in der Anfangszeit des Aufkommens von Hypertext als elektronisch gestütztem Medium sind eine Vielzahl von Navigations- und Orientierungsproblemen identifiziert worden, die besonders beim Umgang mit Hypermedien aber auch mit elektronischen Medien allgemein auftreten [Conklin 1987; Carmel, Crawford & Chen 1992]. Als Reaktion auf diese Problematik wurden mit der Zeit eine Reihe von Navigationsfunktionen und Orientierungsunterstützungsfunktionen entwickelt. Die erfolgreichsten dieser Funktionen sind heute in jedem Web-Browser zu finden und sogar Teil der Definition des Begriffes Hypermedia geworden [Bieber 2000].[26]

Hypermediale Navigation ist daher wohl derzeit der Navigationsansatz, der Browsing generell am besten unterstützt. Probleme beim Einsatz dieses Navigationskonzeptes ergeben sich allerdings aus der Tatsache, dass hypermediale Navigation hauptsächlich auf das Medium Hypertext zugeschnitten ist. Die daraus resultierenden Probleme werden im folgenden Abschnitt diskutiert.

### 4.3.3 Übertragbarkeit hypermedialer Navigation auf Vorlesungsaufzeichnungen

Heutige Hypertext-Browser bieten, wie in 4.4 näher beschrieben, eine Reihe von Möglichkeiten, um Navigation und Orientierung in Hypertext-Dokumenten zu vereinfachen. Im Einzelnen sind dies Volltextsuche, Backtracking, Bookmarks, Breadcrumbs (auch Footprints genannt) und strukturelle Elemente.[27] Die Implementation dieser Navigationshilfen ist sehr stark an das Medium Hypertext, bzw. Mischungen aus Hypertext und Bildern angepasst. Ein Problem bei der Übertragung der den Navigationshilfen zugrunde liegenden Konzepte ergibt sich daher daraus, dass zwischen Text und Vorlesungsaufzeichnungen substantielle Unterschiede bestehen. Text ist, auch wenn er mit Bildern angereichert ist, ein statisches und rein visuell zu erfassendes Medium. In ihrer hergebrachten Form sind hypermediale Navigationshilfen daher auf Text und Bildmaterial anwendbar, nicht jedoch auf Video- und Audiomaterial.

Eine Übertragbarkeit hypermedialer Navigation ist aus diesem Grund nur dann gegeben, wenn die den einzelnen Navigationshilfen zugrunde liegenden Konzepte vom Medium isoliert betrachtet werden können. Um dieser Tatsache Rechnung zu tragen, werden im folgenden Abschnitt zunächst die Charakteristika statischer Hypermedien betrachtet, die sich grundlegend auf die Funktionsweise hypermedialer Navigationselemente auswirken.

### 4.3.4 Charakteristika statischer Hypermedien

Herkömmliche Navigationshilfen für Hypertext sind für statische visuelle Inhalte optimiert. Im folgenden Abschnitt werden daher Unterschiede bei der menschlichen Wahrnehmung statischer visueller Inhalte einerseits und dynamischer visueller und auditiver Inhalte andererseits diskutiert. Die meisten der gemachten Beobachtungen erscheinen auf den ersten

---

durch die Verwendung externer Links [Mertens, Ickerott, Witte & Vornberger 2005] kann jedoch die Bookmarkverwaltung eines Webbrowsers genutzt werden.

[26] Ein großflächiger Überblick zu verschiedenen Definitionen des Begriffs Hypermedia, der diese Entwicklung verdeutlicht, ist in [Tolhurst 1995] zu finden.

[27] Die Kenntnis der Bedeutung der verwendeten Begriffe wird an dieser Stelle vorausgesetzt. Zur näheren Erläuterung der Arbeitsweisen der Navigationshilfen sei jedoch auf [Bieber 2000] verwiesen.

Blick trivial. Bei genauerer Betrachtung stellt das Fehlen der beschriebenen Eigenschaften in zeitabhängigen Medien jedoch das eigentliche Hindernis bei der Anpassung hypermedialer Navigation an Vorlesungsaufzeichnungen dar.

#### 4.3.4.1 Erkennen visuell statischer Inhalte auf einen Blick

*Ein Bild sagt mehr als tausend Worte*. Umformuliert besagt diese Redensart, dass visuelle Eindrücke als Momentaufnahme in Sekundenbruchteilen große Mengen an Information transportieren können. So können beispielsweise zwei verschiedene Texte allein aufgrund ihres Erscheinungsbildes ohne genauere Kenntnis der betreffenden Inhalte verglichen werden. [de Bruijn & Spence 2000] untersuchen die Auswirkungen dieses Sachverhalts auf Suche und Browsing in Text- und statischen Bildmedien. Bei der Navigation in Hypertext-Umgebungen können Leser so auf einen Blick, einen groben Eindruck über den Inhalt und Aufbau einer Seite gewinnen. Backtracking zu einer falschen Seite kann sehr schnell als solches erkannt werden und ähnlich benannte Bookmarks können durch kurzes Annavigieren leicht unterschieden werden. Auf der Zeilenebene können Nutzer dadurch, dass sie Texte schnell überfliegen können sozusagen auf Sicht Scrollen.
Zeitabhängige Medien wie Filme oder Audiodateien geben die in ihnen enthaltene Information im Gegensatz dazu erst beim kompletten Abspielen frei. Diese Tatsache stellt ein erhebliches Hindernis bei der Navigation in derartigen Medien dar [Ponceleon & Dieberger 2001]. Entwicklungen wie der *SpeechSkimmer*, der eine Reihe von Ansätzen zum beschleunigten Abspielen von Sprachaufnahmen implementiert [Arons 1993] oder das im Rahmen von AOF entwickelte Verfahren *random visible scrolling*, das ein Vor- und Zurückspulen von Vorlesungsaufzeichnungen mit variablen Geschwindigkeiten ermöglicht (siehe 4.2.1.2), gehen dieses Problem an, bieten jedoch keine Möglichkeit zur Visualisierung und damit zur unmittelbaren Erfassbarkeit der Inhalte. Selbst unter Einbeziehung derartiger Techniken ist eine Übertragbarkeit hypermedialer Navigationsfunktionen ohne Anpassungen, die den Mangel der Erkennbarkeit visuell statischer Inhalte auf einen Blick ausgleichen, jedoch nicht sinnvoll. Die Anpassung von Navigationsfunktionen, die auf diesem Phänomen basieren ist damit unumgänglich.

#### 4.3.4.2 Klare Abgrenzung von Hypertext-Knoten

Beim Lesen von Texten und bei der Betrachtung visuell statischer Inhalte wird die Lesegeschwindigkeit allein vom Leser bestimmt. So entscheidet beispielsweise beim Lesen von Printmedien wie Büchern oder Zeitungen allein der Leser, wann er eine Seite umblättert oder einen neuen Absatz liest. Diese Feststellung gilt auch für visuell statische Hypermedien. Der Zeitpunkt, zu dem Benutzer von einem Hypertext-Knoten zu einem anderen wechseln, wird vom Benutzer durch eine Navigationsaktion wie das Klicken eines Links gesteuert. Im Vergleich dazu werden beim Abspielen einer Vorlesungsaufzeichnung mit synchronisierten Folien mehrere Knoten hintereinander durchlaufen. Der Übergang von einem Knoten zum anderen ist in derartigen Medien fließend und wird nicht vom Nutzer, sondern vom Medium gesteuert. Hypertext-Knoten sind dagegen klar erkennbar voneinander abgegrenzt. Links und Footprints können daher auf klar definierte Anschnitte des Dokuments verweisen und haben so auch eine konkrete Bedeutung. Zwar können Links auch in audio- oder videobasierten Medien auf eine spezifische Position innerhalb des betreffenden Medienstroms verweisen, durch das selbstständige Abspielen des Mediums können jedoch mehrere Knoten durchlaufen werden, ohne dass dieser Übergang vom Nutzer bemerkt wird. Die Grenzen der betroffenen Knoten verwischen dabei. Einzelne Knoten können daher nur noch schwer als solche identifiziert werden. Eine weitere Schwierigkeit ergibt sich dadurch, dass auch ein einzelner Knoten durch einen Abschnitt der Film- oder Audiospur repräsentiert wird. Besonders relevant ist diese Tatsache für die Implementation von Footprints (siehe 4.4.2.3).

Je nach Implementation kann ein Footprint einen Abschnitt als besucht klassifizieren, obwohl der Benutzer nur einen unwichtigen Teilausschnitt gesehen hat, oder als nicht besucht, obwohl der Nutzer alle relevanten Abschnitte des Knotens bereits gesehen hat.

#### 4.3.4.3 Visualisierung der Dokumentstruktur

Die Struktur textueller Dokumente wird durch verschiedene Bestandteile des Dokumentes wie Überschriften oder hervorgehobenen Text visualisiert. Für Online-Inhalte haben sich derart hervorstechende Elemente sogar als wichtiger erwiesen als für konventionelle, nicht-elektronische Text-Dokumente, da Leser diese Elemente dort vermehrt zur schnellen Orientierung einsetzen [Morkes & Nilsen 1997]. Zur Navigation in textuellen Dokumenten sind derartige Elemente nutzbar, indem der Leser direkt zu der entsprechenden Passage springt. Bei zeitbasierten Medien ist eine Verlinkung von strukturvisualisierenden Elementen in den Film nötig (vgl. dazu 4.5.1.2). Im Umfeld von Vorlesungsaufzeichnungen haben [He, Grudin & Gupta 2000] eine Studie durchgeführt, die die Relevanz aussagekräftiger Folientitel zeigt. Im Zuge dieser Studie raten sie Autoren von Vorlesungsaufzeichnungen dazu, Folien auf einen Schlüsselpunkt zu beschränken, um später Nutzern der Aufzeichnung die Suche und Navigation innerhalb der Aufzeichnung zu erleichtern. Bei genauerer Betrachtung stellt dieses Vorgehen eine Verschiebung der Feinstruktur des Dokumentes auf eine höhere, eigentlich der Grobstruktur vorbehaltene Ebene dar. Ziel dabei ist es, den Mangel eines schnellen und direkten Zugriffs auf die Feinstruktur des Dokumentes wie hervorgehobenen Text oder einzelne Punkte von Bullet-Listen auszugleichen. Nachteil dieser Strategie ist neben dem Verlust wertvoller Strukturinformation ein unnötiges Aufblähen der resultierenden Folienpräsentationen. Idealerweise sollte ein Vorlesungsaufzeichnungssystem daher eher die Möglichkeit bieten entsprechende Feinstrukturen sichtbar und für Navigationszwecke nutzbar zu machen.

### 4.3.5 Fazit

Prinzipiell eignet sich hypermediale Navigation sehr gut für den Einsatz in Vorlesungsaufzeichnungen. Hypermediale Navigationskonzepte erlauben verstärkt die Nutzung vorhandener Dokumentstrukturen und unterstützen Browsing und Suche in besonderem Maße. Ein Nachteil hypermedialer Navigationskonzepte ergibt sich aus der Tatsache, dass hypermediale Navigation in ihrer heutigen Form speziell für Hypertexte und mit Bildern angereicherte Hypertexte entwickelt wurde. 4.3.4 beschreibt die grundlegenden Unterschiede zwischen visuell statischen Medien und zeitabhängigen Medien. Bei der Analyse dieser Unterschiede sind einige Punkte zusammen getragen worden, die bei der Anpassung hypermedialer Navigationskonzepte an Vorlesungsaufzeichnungen beachtet werden müssen. Die teilweise Strukturierung von Vorlesungsaufzeichnungen durch Vortragsfolien bietet in einigen Bereichen Hilfen, dazu ist allerdings eine Verbindung von Folienstruktur und den entsprechenden Film- bzw. Audioabschnitten zu realisieren. In anderen Bereichen, wie beispielsweise bei der in 4.3.4.2 beschriebenen Problematik bei der Implementation von Footprints, sind weitere Anpassungen vorzunehmen. Eine genauere Diskussion der Anforderungen an hypermediale Navigationselemente in Vorlesungsaufzeichnungen unter Berücksichtigung der hier gemachten Beobachtungen erfolgt in 4.5. Vorbereitend wird dazu im folgenden Abschnitt zunächst die konkrete Realisierung hypermedialer Navigation in hypertextuellen Medien betrachtet. Ein besonderer Fokus liegt dabei auf der Einbeziehung der diesen Navigationshilfen zugrunde liegenden Konzepte, um so eine Übertragung dieser Konzepte auf das Anwendungsgebiet Vorlesungsaufzeichnungen zu ermöglichen.

## 4.4 Hypermediale Navigation in Hypertexten

Bereits im Jahr 1945 hat der amerikanische Wissenschaftler Vannevar Bush seine Vision einer Maschine beschrieben, mit der Informationen auf Knopfdruck dynamisch miteinander assoziiert werden können [Bush 1945]. Später sollten die so entstandenen Verknüpfungen in der fiktiven Maschine, der Bush den Namen Memex gab, zur assoziativ geleiteten Informationssuche genutzt werden können. Im Laufe der Jahre entwickelten sich aus diesem Grundgedanken zunächst mehrere frühe Hypertext-Systeme. Der Kerngedanke von Hypertext ist die Verbindung von abgeschlossenen Informationseinheiten, die in diesem Kontext *Knoten* genannt werden, durch direkte Verbindungen, die als *Links* bezeichnet werden. Knoten können durchaus mehrere andere Knoten enthalten und werden in diesem Fall als *Komposit-Knoten* bezeichnet [Bieber 2000]. Links können entweder an den Anfang eines Knotens verweisen oder auf so genannte *Ankerpunkte* innerhalb eines Knotens. Wird ein Knoten über einen Ankerpunkt angewählt, wird der komplette Knoten angezeigt (im Browser als Seite geladen), bei der Darstellung wird jedoch zunächst um den Ankerpunkt herum fokussiert. Über diesen Kerngedanken hinaus beschreiben moderne Definitionen des Begriffes Hypertext jedoch weit mehr als nur ein Gebilde aus Links und Knoten.
Wie bereits in 4.3.2 dargestellt, kristallisierte sich mit dem Aufkommen von Hypertext-Systemen vor allem Orientierungsverlust bei der Navigation in Hypertext-Dokumenten als Problem beim Umgang mit dem Medium heraus.[28] Die im Laufe der Folgejahre entwickelten Navigationsunterstützungskonzepte sind heutzutage in jeden Webbrowser integriert und haben daher einen dermaßen hohen Grad an Transparenz erreicht, dass kaum ein Benutzer die jeweiligen Konzepte wie beispielsweise Footprints oder Backtracking beim Namen nennen kann (vgl. dazu die Einleitung zu Kapitel 5).[29] Hypermedien sind daher weitaus mehr als nur verlinkte Dokumente. Da für die Implementation eines hypermedialen Navigationskonzeptes für Vorlesungsaufzeichnungen ein tieferes Verständnis hypermedialer Navigation unumgänglich ist, werden in diesem Abschnitt alle hypermedialen Navigationselemente, die Bestandteil moderner Definitionen des Begriffes Hypermedia wie beispielsweise in [Bieber 2000] sind, eingehender diskutiert und analysiert. Aufgrund der bereits genannten Transparenz hypermedialer Navigationselemente im Alltagsgebrauch mag die ausführliche Darstellung auf den ersten Blick trivial erscheinen. Gerade wegen dieses eher unbewussten Umgangs mit hypermedialen Navigationselementen ist allerdings eine klare und detaillierte Beschreibung wichtig. Durch diese Beschreibung werden konzeptuelle Herausforderungen bei der Übertragung hypermedialer Navigationshilfen auf Vorlesungsaufzeichnungen deutlich. Darüber hinaus wird eine Reihe von Punkten herausgearbeitet, die bei der Umsetzung hypermedialer Navigationshilfen beachtet werden müssen. Da moderne Webbrowser das wohl bekannteste Beispiel für eine gelungene Implementation hypermedialer Navigationshilfen geben, geht die nachfolgende Beschreibung zunächst von der Realisierung hypermedialer Navigation im Microsoft Internet Explorer als einem typischen Webbrowser aus. Anschließend werden die Aufgaben hypermedialer Navigationselemente bei der Arbeit mit hypermedialen Dokumenten am Beispiel hypertextueller Dokumente betrachtet.

### 4.4.1 Hypermediale Navigationselemente in modernen Webbrowsern

Laut [Bieber 00] bilden Backtracking, Bookmarks, Footprint, strukturelle Elemente und Volltextsuche einen essentiellen Bestandteil hypermedialer Umgebungen. Unter leicht variierenden Namen finden sich diese Navigationshilfen in nahezu identischer Realisierung

[28] Ein Überblick über Hypertext-Anwendungen aus der Zeit vor der Entwicklung des Internet wird in [Conklin 1987] gegeben.
[29] Bookmarks und Volltextsuche bilden hier mit Sicherheit bei den meisten Benutzern eine Ausnahme.

in allen heutzutage verbreiteten Webbrowsern. Der Netscape Navigator, der Microsoft Internet Explorer, der Safari Webbrowser von Apple, der Mozilla Firefox oder auch der Opera Webbrowser verfügen über dieselben Navigationselemente. Abb. 4-2 zeigt die Realisierung der Elemente im Microsoft Internet Explorer.

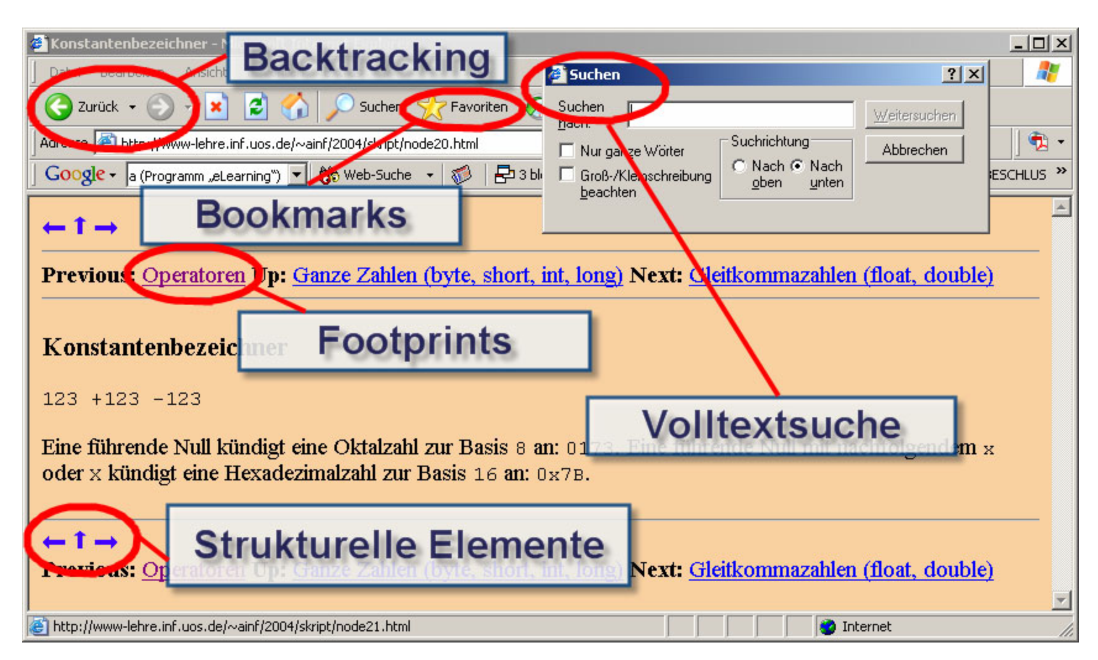



**Abb. 4-2: Navigationsunterstützung für Hypertext-Dokumente im Internet Explorer**

Vor Hintergrund der bereits erwähnten Transparenz hypermedialer Navigationshilfen ist vor allem die Tatsache zu erwähnen, dass Footprints und strukturelle Elemente dokument- und interaktionsvariante Navigationshilfen sind und daher eher mit dem Dokument verschmelzen. Sie sind demzufolge im Gegensatz zu Bookmarks und Backtracking nicht an festen Positionen im Bedienmenü des Browsers zu finden. Die Volltextsuche über das aktuelle Dokument ist in modernen Webbrowsern über eine Tastenkombination und einen Menüpunkt zu erreichen oder wie beispielsweise im Mozilla Firefox direkt in die Browseroberfläche integriert.

## 4.4.2 Hypermediale Navigationselemente und deren Aufgaben

Backtracking-Listen, Bookmarks und Footprints (auch Breadcrumbs genannt) bieten Nutzern eine individualisierte Zusammenstellung ausgeführter Navigationsaktionen [Nielsen 1990b]. Sie bilden damit kontextuelle Orientierungspunkte die zusammen mit der Struktur des jeweiligen Hypertext-Dokuments die kognitive Belastung bei der Suche und Navigation in hypertextuellen Umgebungen verringern und so die Nutzbarkeit von Hypertext-Systemen entscheidend verbessern [Carmel, Crawford & Chen 1992]. Die jeweiligen Navigationselemente erfüllen dabei unterschiedliche Aufgaben und basieren auf verschiedenen Konzepten. Nichtsdestotrotz sind die jeweiligen Navigationselemente nicht nur einzeln zu betrachten sondern bilden in modernen Webbrowsern ein auf Synergie-Effekten beruhendes Konzept. Im weiteren Verlauf werden zunächst die Navigationselemente hypertextueller Umgebungen nach [Bieber 2000] einzeln beschrieben und bezüglich ihrer Funktion analysiert. Daran anschließend folgt eine kurze Betrachtung des Zusammenspiels der Navigationselemente.

### 4.4.2.1 Volltextsuche

Von [Morkes & Nielsen 1997] wird Volltextsuche als einer der wichtigsten Suchmechanismen in Hypertext-Systemen gewertet. Volltextsuche erlaubt einen direkten Zugriff auf

beliebige Teile eines Dokumentes. Die Suche ist damit strukturunabhängig und erlaubt eine schnelle Orientierung besonders in größeren Dokumenten. Eine rudimentäre Volltextsuche ist in alle herkömmlichen Webbrowser über ein Popup-Fenster wie in Abb. 4-3 gezeigt, integriert. Eine derartige Suche hat den Nachteil, dass sie nur die augenblicklich gezeigte Webseite abdeckt.

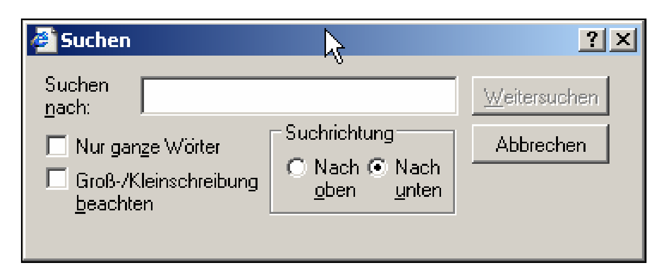



**Abb. 4-3: Volltextsuche im Microsoft Internet Explorer**

Volltextsuche über mehrere (unter Umständen miteinander verlinkte) Webseiten, die zu einem Dokument oder einem Webauftritt wie beispielsweise der Webseite einer Universität oder einer Firma gehören, ist über die Einbindung von Suchfunktionen auf den entsprechenden Seiten möglich. Darüber hinaus ist in vielen Browsern ein separates Eingabefeld zur Suche im WWW vorhanden.

### 4.4.2.2 Strukturelle Elemente

Zu den strukturellen Elementen hypertextueller Dokumente sind zunächst all die Elemente zu zählen, die auch in einem konventionellen Textdokument als strukturelle Elemente betrachtet werden können. Überschriften die in einem herkömmlichen Textdokument einen neuen Abschnitt markieren, kennzeichnen in einem hypertextuellen Äquivalent zu diesem Textdokument in den meisten Fällen den Beginn eines neuen Knotens. Gut strukturierte Hypertextdokumente verfügen darüber hinaus über verlinkte Inhaltsverzeichnisse und *next, up, previous*-Links wie in Abb. 4-4 dargestellt.

**Previous:** Operatoren **Up:** Ganze Zahlen (byte, short, int, long) **Next:** Gleitkommazahlen (float, double)

**Abb. 4-4: „next, up, previous"-Links in einem Hypertext-Dokument**[30]

Im Kontext von Vorlesungsaufzeichnungen empfiehlt [Brusilovsky 2000] den Einsatz von *next, up, previous*-Links als ein einfach zu realisierendes Mittel zur Navigation. Diese Form struktureller Elemente basiert dabei auf der Knotenstruktur eines hypermedialen Dokumentes, da sie eine direkte Navigation zu benachbarten Knoten ermöglicht.
Wie bereits in 4.3.4.3 erwähnt, haben [Morkes & Nilsen 1997] die Bedeutung von visuell hervorgehobenen strukturellen Elementen wie Überschriften, kursiv geschriebenen Wörtern oder Stichpunkten in Bullet-Listen für die Orientierung in hypertextuellen Dokumenten gezeigt. In Verbindung mit der Möglichkeit, in einem Textdokument Abschnitte zu überfliegen und quasi auf Sicht zu scrollen (vgl. dazu 4.3.4.1), erlauben strukturelle Elemente in Hypertext-Dokumenten das direkte Auffinden und Ansteuern von gesuchten Passagen.

### 4.4.2.3 Footprints

Unter Footprints oder auch Breadcrumbs versteht man jede Art von visuellen Indikatoren, die anzeigen, ob ein Knoten bereits besucht wurde oder ein Link bereits genutzt wurde

[30] Das Dokument wurde mit dem Dokumentationsgenerator mas2text [Herzog, Thiesing & Vornberger 1999] generiert, der erzeugte Hypertext-Dokumente automatisch mit diesen strukturellen Elementen versieht.

[Parmar 1993]. [Tauscher 1996] findet mit dem Begriff *“Already-visited”-cues* eine treffende Umschreibung und liefert einen kurzen Überblick über verschiedene Implementationen. Die Aufgabe von Footprints besteht darin, dass sie eine einfache Unterscheidung zwischen Webseiten oder Teilen eines hypermedialen Dokumentes ermöglichen, die der Nutzer bereits besucht hat und solchen, die für den Nutzer neu sind.
In graphischen Webbrowsern werden Footprints durch Einfärbung von Links realisiert [Tauscher 1996]. So zeigt beispielsweise die Einfärbung des Link-Textes „Operatoren“ in Abb. 4-5, dass der durch den Link referenzierte Knoten bereits besucht wurde.

**Previous:** Operatoren **Up:** Ganze Zahlen (byte, short, int, long)

**Abb. 4-5: Footprints in einem Hypertext-Dokument**

Footprints sind in hypertextuellen Dokumenten vor allem aufgrund der in 4.3.4.2 dargestellten Abgrenzung von Hypertext-Knoten effektiv. Dadurch, dass ein Link genau einen Knoten referenziert, kann ein Footprint auch exakt einem Knoten zugeordnet werden. Aufgrund ihrer Bindung an hypertextuelle Knoten und der Tatsache, dass die Grobstruktur eines Dokumentes in den meisten Fällen sehr stark mit der Knotenstruktur korreliert, sind Footprints zudem an die Grobstruktur des Dokumentes gebunden (vgl. dazu auch die in 4.4.2.2 beschriebene Verbindung von *next, up, previous*-Links mit der Knotenstruktur eines Hypertextes)

#### 4.4.2.4 Backtracking

Bereits in sehr frühen Systemen hat sich Backtracking als sinnvolle Methode erwiesen, um Orientierungsproblemen in hypertextuellen Systemen zu begegnen [Nielsen 1990b]. Durch Betätigen des Backtracking-Knopfes können Benutzer zu der Stelle zurückkehren, die sie durch Anklicken eines irreführenden Links oder durch eine Sequenz hintereinander ausgeführter nicht zielführender Navigationsaktionen verlassen haben. Backtracking-Listen ermöglichen dabei das Überspringen von Backtracking-Punkten und erlauben es so, mehrere Schritte auf einmal zurückzugehen. In modernen Webbrowsern wird Backtracking in der Regel durch zwei Stacks realisiert, wie in [Cockburn, McKenzie & JasonSmith 2002] beschrieben. Auf einem der beiden Stacks, dem Back-Stack, werden besuchte Knoten in der Reihenfolge der Aufrufe gespeichert. Wird eine Backtracking-Aktion ausgeführt, wird der oberste Eintrag vom Back-Stack entfernt und in den zweiten Stack, den Forward-Stack, eingefügt. Durch Betätigen des Forward-Knopfes kann eine Backtracking-Aktion, mit der der Nutzer zu weit zurückgegangen ist, zurückgenommen werden.[31] In modernen Webbrowsern sind beide Knöpfe in der Regel direkt nebeneinander angeordnet, wie in Abb. 4-6 dargestellt.




**Abb. 4-6: Backtracking Bedienelemente im Internet Explorer**

Backtracking hat den Vorteil, dass es auch bei falscher Interpretation der Funktionsweise intuitiv einfach zu bedienen ist und dass die zugehörigen Bedienelemente sehr platzsparend angeordnet werden können [Greenberg & Cockburn 1999]. Backtracking ist damit eines der wichtigsten hypermedialen Navigationselemente [Cockburn, McKenzie & JasonSmith 2002]

[31] Der oberste Eintrag des Forward-Stacks wird dabei wieder auf den Back-Stack verschoben. Eine Diskussion der Vor- und Nachteile dieser Funktionsweise sowie die Darstellung eines alternativen zeitbasierten Backtracking ist in [Cockburn, McKenzie & JasonSmith 2002] zu finden.

und hat sich in mehreren Studien als eines der am häufigsten genutzten Navigationsmittel beim Browsen überhaupt gezeigt [Aula, Jhaveri & Käki 2005].[32]
[Greenberg & Cockburn 1999] beschreiben die Arbeit mit Backtracking als eine „'click until the desired page is recognized' strategy". Backtracking funktioniert in hypertextuellen Medien damit vor allem aufgrund der raschen Erkennbarkeit visuell statischer Inhalte (vgl. 4.3.4.1). Durch die in 4.3.4.2 beschriebene klare Abgrenzung von Hypertext-Knoten kann jeder Durchführung einer Navigationsaktion genau ein Knoten als aktueller Knoten zugeordnet werden. Dieser Knoten wird jeweils in den Back-Stack eingefügt und dient später als Ansprungpunkt bei Zurücknahme der betreffenden Navigationsaktion.

#### 4.4.2.5 Bookmarks

Bookmarks sind vom Nutzer selbst zusammengestellte Linksammlungen, die auf verschiedene Hypertext-Dokumente oder Knoten innerhalb eines hypertextuellen Dokumentes verweisen. Zur Erstellung eines Bookmarks können beliebige Knoten innerhalb von Hypertext-Dokumenten verwendet werden. Zu späteren Zeitpunkten können die Bookmarks dazu verwendet werden, die jeweiligen Knoten durch einen Mausklick in einer Bookmark-Liste, wie in Abb. 4-7 gezeigt, direkt anzusteuern.

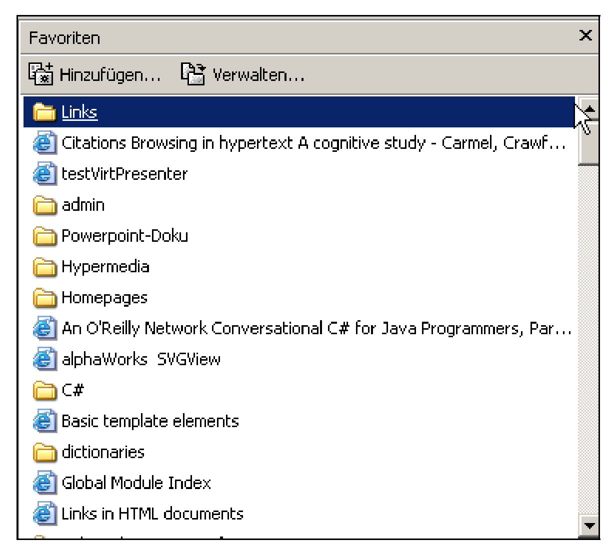



**Abb. 4-7: Bookmark-Liste im Internet-Explorer**

Bookmarks werden von erfahrenen Nutzern vergleichsweise oft eingesetzt [Aula, Jhaveri & Käki 2005] und stellen damit ein wichtiges Navigationselement in hypertextuellen Umgebungen dar. Wie bereits in 4.3.4.1 erwähnt, basiert die Arbeit mit Bookmarks oft auf der Unterscheidbarkeit visuell statischer Inhalte auf einen Blick. Die technische Realisierung von Bookmarks in Hypertext ist auf der in 4.3.4.2 dargestellten klaren Abgrenzung von Hypertextknoten aufgebaut. Wird ein Bookmark erzeugt, zeigt es auf den zum Zeitpunkt seiner Erstellung aktuellen Hypertext-Knoten.

[32] Ein eher anekdotischer Beleg für die Relevanz von Backtracking sind Strategien beim Umgang mit bestimmten Webseiten, wie der Lernplattform Stud.IP an der Universität Osnabrück oder Suchmaschinen. Bei Stud.IP ist aufgrund der Implementation die Verwendung des Back-Knopfes nicht möglich. Erfahrene Benutzer öffnen Links daher oft in neuen Fenstern oder Tabs um zum Ausgangspunkt zurückkehren zu können. Auch bei der Arbeit mit Suchmaschinen ist in [Aula, Jhaveri & Käki 2005] eine vermehrte Nutzung von Tabs und mehreren gleichzeitig geöffneten Browserfenstern festgestellt worden. Durch das Öffnen von Suchergebnissen in neuen Fenstern oder Tabs kann hier die Wartezeit beim Backtracking durch ein erneutes Absenden der Suchanfrage vermieden werden.

### 4.4.3 Zusammenspiel der Navigationselemente

Bei der Navigation in Hypertext werden die oben beschriebenen Navigationselemente in der Regel nicht unabhängig voneinander, sondern als Teile eines größeren Navigationskonzeptes verwendet. So werden beispielsweise strukturelle Elemente wie ein verlinktes Inhaltsverzeichnis durch Footprint-Eigenschaften der Links modifiziert. Durch Backtracking kann eine Reihe mit Footprints markierter Links schnell hintereinander begutachtet werden. Wäre Backtracking in diesem Fall nicht möglich, müsste die Seite, auf der die entsprechenden Links stehen jedes einzelne Mal wieder neu gesucht werden.[33] Auch die Verwendung von Bookmarks als Ausgangspunkt für die weitere Suche ist ein Beispiel für die Verbindung der Nutzung verschiedener der oben beschriebenen Navigationshilfen. Beim Einsatz der Textsuche über mehrere Knoten in einem hypertextuellen Dokument, wie beispielsweise mit einer Suchmaschine möglich, erleichtern Footprints die Unterscheidung von Suchergebnissen als bekannte und unbekannte Knoten. Durch Backtracking kann die Verwendung eines strukturellen Elementes zurückgenommen werden, um wieder zum Ausgangspunkt zurück zu gelangen.

Bei der Beobachtung der täglichen Arbeit mit Hypertext lassen sich noch viele weitere Beispiele finden, die das Zusammenspiel der oben beschriebenen hypertextuellen Navigationselemente illustrieren. Im Rahmen der vorliegenden Arbeit ist dabei vor allem die Tatsache relevant, dass die Arbeitsweise von Benutzern bei der Navigation in Hypertexten den Einsatz von hypermedialen Navigationselementen in (fast) beliebigen Kombinationen miteinander verknüpft.

### 4.4.4 Zusammenfassung

Die im Laufe der Zeit in jedem Webbrowser zum Standard gewordenen hypermedialen Navigationselemente Volltextsuche, Bookmarks, Footprints, Backtracking und strukturelle Elemente basieren zu einem Großteil auf den in 4.3.4 dargestellten Charakteristika hypertextueller Medien. Darüber hinaus sind hypermediale Navigationshilfen nicht als eine Ansammlung von unabhängigen Navigationskonstrukten zu verstehen sondern als eine zusammenhängende Implementation aufeinander abgestimmter Navigationsfunktionalitäten.

Bei der Umsetzung hypermedialer Navigation für Vorlesungsaufzeichnungen sind diese Punkte zu berücksichtigen, da sonst die Möglichkeiten hypermedialer Navigation nur teilweise ausgeschöpft werden und synergetische Effekte bei der Kombination der Navigationselemente verloren gehen würden.

## 4.5 Anforderungen an hypermediale Navigationskonzepte für Vorlesungsaufzeichnungen

Ein Folienvortrag verfügt durch die Aufteilung thematischer Blöcke in Folien bereits über eine durchgehende Strukturierung. Bei vielen Dozenten ist durch die Animation der Folien sogar eine Vorgabe der Feinstruktur des Vortrags gegeben, da einzelnen Animationen in vielen Fällen ein Unterthema des in der Folie behandelten Themas zugeordnet ist. Diese Struktur kann zur Transformation der Aufzeichnung in ein Hypermedium genutzt werden, indem die Zeitintervalle zwischen zwei Folienwechseln oder Folienanimationen zusammen mit dem animierten Folienbestandteil bzw. der betreffenden Folie als Knoten eines hypermedialen Dokumentes verstanden werden. [Graca Pimentel, Abowd & Ishiguro 2000] beschreibt in diesem Sinne die automatische Erstellung von Hypertext-Dokumenten durch die Aufzeichnung von Vorlesungen auf Folienebene (vgl. dazu auch 5.5).

[33] Alternativ könnte die Seite auch als Bookmark gespeichert werden oder es könnte eine der in Fußnote 32 auf Seite 71 beschriebenen Ausweichstrategien verwendet werden. Der Einsatz von Bookmarks würde allerdings auch den Rückgriff auf ein anderes hypermediales Navigationselement bedeuten.

Die Anpassung der in 4.4.2 beschriebenen Navigationselemente an die Nutzung in hypermedial aufbereiteten Vorlesungsaufzeichnungen gestaltet sich im Gegensatz zu einer reinen Transformation der Aufzeichnung in eine Ansammlung miteinander verlinkter Knoten vor allem in Anbetracht der in 4.4.2.1 bis 4.4.2.5 erläuterten Bindung der jeweiligen Navigationselemente an statische Hypermedien etwas schwieriger. Im Folgenden werden zunächst Anforderungen an die Anpassung der jeweiligen Navigationselemente herausgearbeitet. Daran anschließend wird aus diesen Anforderungen eine Reihe von Eigenschaften der Viewer-Komponente abgeleitet, die zur Umsetzung hypermedialer Navigation in Vorlesungsaufzeichnungen erfüllt sein müssen.

### 4.5.1 Navigationselemente

Durch die Kopplung von Folien und Aufzeichnung ist in einer Vorlesungsaufzeichnung jeder Abschnitt der Video-, bzw. Audioaufzeichnung einer Folie oder einer Folienanimation zugeordnet. In einer didaktisch sinnvollen Präsentation der Folien kann dabei davon ausgegangen werden, dass die entsprechende Folie bzw. die entsprechende Folienanimation eine prägnante Beschreibung des im jeweiligen Video- oder Audioabschnitt behandelten thematischen Punktes darstellt.[34] Durch eine Thumbnail-Übersicht der im Vortrag verwendeten Folien ist daher ein visuell statischer Überblick über Schlüsselszenen der Aufzeichnung möglich. Derartige Folien-Übersichten werden derzeit in den meisten Vorlesungsaufzeichnungssystemen eingesetzt und bieten dort die Möglichkeit jeweils zum Anfangspunkt der Präsentation der jeweiligen Folie zu navigieren [Hürst & Götz 2004]. Im Rahmen der Entwicklung eines hypermedialen Navigationskonzeptes für Vorlesungsaufzeichnungen kann ein solcher, visuell statischer Überblick weitaus umfassender genutzt werden. Durch die Kopplung von Folien an definite Filmsequenzen können die jeweiligen Folien mit Abschnitten der Zeitleiste assoziiert werden. Bei Einbindung von animierten Folienbestandteilen nach dem *linking by interacting* Paradigma (vgl. 4.2.1.3) kann sogar eine Verbindung zwischen der Zeitleiste und einzelnen Stichpunkten oder Teilen einer Grafik hergestellt werden. Auf dieser Grundlage ist es möglich, die Struktur und sogar die Feinstruktur eines Vortrags bidirektional mit der Zeitleiste zu verbinden. Punkte oder Abschnitte der Zeitleiste können zur Visualisierung der zugehörigen Strukturinformation genutzt werden und Bestandteile des strukturellen Vortragsüberblicks können zur Anzeige der assoziierten Abschnitte auf der Zeitachse verwendet werden. Durch eine solche Verbindung von Zeit und Struktur kann das Fehlen der in 4.3.4 erörterten Eigenschaften visuell statischer Hypermedien zumindest teilweise kompensiert werden. Dies ermöglicht bei der Anpassung hypermedialer Navigationsfunktionalitäten an Vorlesungsaufzeichnungen in vielen Punkten einen Rückgriff auf einen visuell statischen Überblick. In anderen Punkten sind jedoch auch grundlegend zeitbasierte Aspekte von Vorlesungsaufzeichnungen zu berücksichtigen. Im Folgenden wird unter Berücksichtigung dieser beiden Tatsachen diskutiert, welche Anforderungen an die Umsetzung der einzelnen Navigationshilfen zu stellen sind. Der Unterpunkt strukturelle Elemente (4.5.1.2) wird dabei im Gegensatz zu den anderen Unterpunkten weitaus näher an der Umsetzung und weniger konzeptuell diskutiert. Der Grund hierfür ist, dass strukturelle Elemente in der einen oder anderen Form bereits in vielen Systemen zu Vorlesungsaufzeichnung umgesetzt wurden (vgl. Kapitel 5). Hier müssen daher keine grundlegend neuen Konzepte entwickelt, sondern bestehende Ansätze modifiziert werden.

[34] Dies ist schließlich gerade der Sinn der Verwendung von Folien. Vgl. dazu auch [He, Grudin & Gupta 2000] aus technischer oder [Weidenmann 2002] aus didaktischer Sicht.

#### 4.5.1.1 Volltext-Suche

Eine Schwierigkeit bei der Umsetzung von Volltextsuche für Vorlesungsaufzeichnungen liegt in der kompletten Indizierung des Materials [Bell, Cockburn, McKenzie & Vargo 2001]. Werden die Folieninhalte symbolisch repräsentiert, ist dies recht einfach zu realisieren [Lauer und Ottmann 2002]. Beim Einsatz von Tafel und Kreide treten eine Reihe von Problemen auf, wie beispielsweise in [Liu & Kender 2004] beschrieben. Neben der Verwendung visueller Medien ist auch die Nutzung der Tonspur einer Aufzeichnung zur Volltextsuche denkbar (vgl. dazu 5.7.2).
Zur effektiven Nutzung von Volltext-Suche in Vorlesungsaufzeichnungen ist die Möglichkeit eines feingranularen Zugriffs auf das aufgezeichnete Material entscheidend [Brusilovsky 2000]. Besonders in Anbetracht der Tatsache, dass gerade auf Folien mit vielen Stichpunkten oft mehrere Minuten Redezeit verwendet werden, ist eine Suchfunktion, die beispielsweise an den Anfangspunkt der betreffenden Folie in der Aufnahme springt nur bedingt sinnvoll.
Ein weiterer nicht zu vernachlässigender Aspekt der Funktionsweise von Volltextsuche ist die Einbindung der Suchbegriffe in Kontexte. Aus diesem Grund arbeiten beispielsweise alle modernen Suchmaschinen mit der Präsentation von umgebenden Textausschnitten, in denen die jeweiligen Suchbegriffe zu finden sind. Bei Vorlesungsaufzeichnungen ist nicht, wie bei visuell statischen Dokumenten, nur die strukturelle Einordnung sondern darüber hinaus auch die zeitliche Einordnung des Vorkommens von Suchbegriffen zu berücksichtigen.

#### 4.5.1.2 Strukturelle Elemente

Wenn Bücher als Lerngrundlage genutzt werden, wird der Inhalt des Buches üblicherweise selektiv und mit Schwerpunkten auf kritischen Teilabschnitten genutzt. Dies gilt in ähnlicher Form für Vorlesungsaufzeichnungen. Wenn sie als Lerngrundlage genutzt werden sollen, müssen Studierende sehr feingranular die Möglichkeit haben, einzelne Teilbereiche gezielt anzusteuern, zu wiederholen oder auch zu überspringen [Hürst & Götz 2004]. Da strukturelle Elemente in hypermedialen Navigationsansätzen neben Bookmarks die einzige semantisch orientierte Navigationshilfe sind, kommt ihnen daher eine besondere Bedeutung zu.[35] Ein weiteres Argument für die besondere Relevanz strukturbasierter Navigation stellt bei webbasierten Systemen, in denen das Filmmaterial gestreamt wird, die Tatsache dar, das jede Navigationsaktion mit einigen Sekunden Wartezeit bis zum weiteren Abspielen des Films ab der annavigierten Position verbunden sein kann.
Strukturelle Elemente in Vorlesungsaufzeichnungssystemen sind entlang der Dimension Grobebene-Feinebene sowie anhand der beiden Merkmale Strukturvisualisierung und Kontextorientierung zu unterscheiden. Die aktuell gegenpolige Einordnung von Strukturvisualisierung und Kontextorientierung ist dabei nur eine De-facto-Kategorisierung, die auf der Verteilung dieser Merkmale bei gegenwärtigen Vorlesungsaufzeichnungssystemen basiert. Systeme, die strukturvisualisierende strukturelle Elemente verwenden, erlauben nicht den Einsatz von kontextorientierten strukturellen Elementen und umgekehrt. Systeme wie eClass (vgl. 5.5) oder auch das von [Liu & Kender 2004] vorgestellte System erlauben zwar das direkte Anklicken von handschriftlichen Annotationen zur Navigation zum Anfangszeitpunkt der Erstellung der jeweiligen Annotation im Film, verfügen aber nicht über ein Analogon zu den in 4.4.2.2 dargestellten *previous, up, next*-Navigationselementen. Andere Systeme wie AOF (siehe 5.2) oder MANIC (siehe 5.1) verfügen über derartige

[35] Auch mit Bookmarks ist es möglich, semantische Überblicke über den Inhalt von Vorlesungsaufzeichnungen zu erstellen. Ein extremes Beispiel hierfür ist das Projekt Knowledgebay, in dem Inhaltsverzeichnisse für Vorlesungsaufzeichnungen im Nachhinein durch Studierende erstellt werden [Sporer, Köstlbacher & Erbacher 2005]. Dies ist jedoch kein typisches Szenario für den Einsatz von Bookmarks. In der Regel werden Bookmarks von Lernern als Form der individuellen Strukturierung des Materials genutzt, wohingegen strukturelle Elemente die vom Dozenten vorgegebene Struktur der Vorlesung widerspiegeln.

Navigationselemente. Diese Systeme bieten jedoch, aufgrund technischer Einschränkungen der zur Darstellung der Übersichtsfolien verwendeten Bildformate, nicht die Möglichkeit, einzelne Objekte oder wie in eClass handschriftliche Annotationen, an deren Erscheinen auf dem Bildschirm der jeweils anzunavigierende Zeitindex gekoppelt ist, anzuklicken. Der Detaillierungsgrad struktureller Navigationselemente ist damit in sehr vielen Systemen auf die Folienebene beschränkt und demzufolge vergleichsweise grob [Hürst & Götz 2004]. Wünschenswert, jedoch kaum implementiert sind hingegen strukturelle Navigationsmöglichkeiten auf Zeilenebene [Brusilovsky 2000].

In gut strukturierten Hypertextdokumenten sind sowohl kontextorientierte strukturelle Navigationselemente wie beispielsweise eine „previous, up, next“-Navigation als auch strukturvisualisierende strukturelle Elemente wie beispielsweise ein hierarchisch strukturiertes Inhaltsverzeichnis zu finden. In Vorlesungsaufzeichnungen ist aufgrund der zeitbasierten Natur des Mediums ein sehr hoher Detaillierungsgrad sowohl kontextorientierter als auch strukturvisualisierender struktureller Elemente essentiell.

Als strukturvisualisierende Elemente eignen sich vor allem Thumbnails. So stellen [Bernstein, Brown, Frisse, Glushko, Landow & Zellweger 1991] im Kontext statischer Hypermedien fest, dass Thumbnails situierte Überblicke bieten und damit eine wertvolle Orientierungshilfe darstellen. Thumbnails erleichtern das Wiedererkennen von bereits Gesehenem und vermindern so den kognitiven Workload [Aula, Jhaveri & Käki 2005].[36] Sind Thumbnails groß genug, kann enthaltener Text gelesen oder Grafiken erkannt werden. In diesem Fall können sie auch zur Orientierung in nicht bekannten Abschnitten einer Aufzeichnung genutzt werden. Ein Problem derzeitiger Vorlesungsaufzeichnungssysteme ist jedoch, dass Thumbnail-Überblicke den Inhalt der Folien so stark verkleinern, dass er kaum noch nutzbar ist [Hürst & Götz 2004].

Als Fazit dieser Betrachtung ist festzuhalten, dass sowohl kontextbasierte als auch strukturvisualisierende strukturelle Navigationselemente jeweils mit sehr hoch auflösendem Detaillierungsgrad zu implementieren sind. Als weitere Anforderung kommt die Erkennbarkeit des Inhaltes von Thumbnail-Bildern hinzu, die im Rahmen struktureller Elemente zur Strukturvisualisierung genutzt werden.

#### 4.5.1.3 Footprints

Wie bereits in 4.3.4.2 erwähnt, basiert die Funktionsweise von Footprints in Hypertext zu einem Großteil auf der Abgrenzung von Hypertext-Knoten. Da eine solche Abgrenzung in zeitabhängigen Hypermedien kaum gegeben ist (vgl. 4.3.4.2), ist eine starke Bindung von Footprints an die Struktur eines hypermedialen Dokumentes, wie sie in visuell statischen Hypermedien gegeben ist (vgl. 4.4.2.3) nicht sinnvoll. Aufgrund dieser Tatsache ist eine strukturbasierte Implementation von Footprints als Navigationshilfe in Vorlesungsaufzeichnung nur bedingt einsatztauglich. Als ein Beispiel hierfür kann die Implementation von Footprints im Kontext des Aufzeichnungssystems eTEACH angeführt werden, deren Vor- und Nachteile in 5.6 diskutiert werden. Das Hauptproblem bei der strukturbasierten Implementation von Footprints in Vorlesungsaufzeichnungssystemen ist, wie bereits in 4.3.4.2 angedeutet, die Schwierigkeit, ein Kriterium zu finden, das besagt, ob ein Knoten vom Nutzer bereits kognitiv verarbeitet wurde oder nur unwesentliche Ausschnitte des dem Knoten zugeordneten Teils der Aufzeichnung abgespielt worden sind. In statischem Hypertext ist ein solches Kriterium durch die Begutachtung des durch das Footprint gekennzeichneten Knoten in Verbindung mit der in 4.3.4.1 diskutierten unmittelbaren Erfassbarkeit visuell statischer Inhalte gegeben (vgl. dazu auch 4.4.2.3).

[36] Voraussetzung sind dabei Unterscheidbarkeit in Farbe oder Layout der jeweils gezeigten Thumbnails oder Lesbarkeit von Titeltexten oder Ähnlichem [Kaasten, Greenberg & Edwards 2002].

Weitaus sinnvoller als der Versuch, Footprints für Vorlesungsaufzeichnungssysteme strukturbasiert zu implementieren ist daher in diesem Zusammenhang die Konzeption eines zeitbasierten Analogons. Da eine zeitbasierte Markierung bereits abgespielter Abschnitte der Aufzeichnung jedoch für sich betrachtet nur wenig aussagekräftig ist, muss neben einer zeitbasierten Markierung auch eine visuelle Kennzeichnung der zu der jeweiligen Markierung gehörigen Struktur des Dokumentes integriert werden.

#### 4.5.1.4 Backtracking

Backtracking dient zur Umkehrung beliebiger Navigationsaktionen. Daher müssen bei der Implementation von Backtracking für Vorlesungsaufzeichnungssysteme sowohl zeitbasierte als auch strukturbasierte Navigationsaktionen berücksichtigt werden. Die technische Realisierung von Backtracking durch die Bindung an Hypertext-Knoten (vgl. 4.4.2.4) ist daher für Vorlesungsaufzeichnungssysteme nicht geeignet. Da strukturbasierte Navigation in Vorlesungsaufzeichnungen aufgrund der Synchronisation struktureller Elemente mit der Videoaufzeichnung immer auch zeitbasierte Navigation ist, erscheint eine zeitbasierte Umsetzung des Konzeptes Backtracking für diesen Anwendungsbereich als weitaus sinnvoller. Ein Eintrag im Backtracking-Stack besteht damit nicht aus einem hypermedialen Knoten sondern einem Zeitindex. Zu berücksichtigen ist hierbei der Wegfall der in 4.3.4.1 beschriebenen Erkennbarkeit visuell statischer Inhalte auf einen Blick und dessen Bedeutung für die Arbeitsweise von Backtracking (vgl. 4.4.2.4). Um zeitbasiertes Backtracking effektiv zu realisieren, ist daher eine enge Verbindung des Backtracking-Navigations-Elementes mit statischen strukturellen Übersichtselementen, wie beispielsweise der in 4.5.1.2 beschriebenen Thumbnail-Übersicht entscheidend. Eine feingranularere Verbindung beispielsweise mit Folienelementen der Thumbnail-Übersicht, die in der Präsentation animiert sind, ist wünschenswert.

#### 4.5.1.5 Bookmarks

Zur Anpassung an Vorlesungsaufzeichnungen müssen Bookmarks zeitbasiert implementiert werden. Dies ist konzeptuell recht einfach zu realisieren, indem Bookmarks an beliebige Zeitpunkte innerhalb einer Aufzeichnung verweisen. Durch die Möglichkeit zur nutzerseitigen Manipulation der Abspieldauer kann die in 4.3.4.2 beschriebene Abgrenzung von hypertextuellen Knoten emuliert werden. Im Gegensatz zu Bookmarks in Hypertext wäre mittels dieser freien Wählbarkeit von Start- und Endpunkten sogar eine Erweiterung des klassischen Bookmark-Konzeptes gegeben. Ein zeitbasiertes Bookmark könnte damit auf einen Ausschnitt aus einem Knoten beschränkt werden. Es könnte aber auch über mehrere Knotengrenzen hinweg faktisch einen neuen Knoten schaffen. Vor allem in Hinblick auf den in 4.4.2.5 dargestellten Zusammenhang der in 4.3.4.1 erörterten Erkennbarkeit visuell statischer Informationen auf einen Blick mit der Arbeitsweise von Bookmarks in klassischem Hypertext ist eine visuelle Einordnung des von einem Bookmark bezeichneten Abschnitts in die strukturelle Übersicht der Aufzeichnung wichtig.

### 4.5.2 Zusammenspiel der Navigationselemente

In den vorangegangenen Betrachtungen zur Adaptierbarkeit der Konzepte, die einzelnen hypermedialen Navigationselementen zugrunde liegen, hat sich gezeigt, dass einige Elemente primär zeitbasiert und andere primär strukturbasiert umgesetzt werden sollten. Darüber hinaus hat die Betrachtung der einzelnen Elemente deutlich gemacht, dass eine Verbindung zeitbasierter und strukturbasierter Navigation für alle einzelnen Elemente sinnvoll ist. Auch in Hinblick auf das in 4.4.3 diskutierte Zusammenspiel hypermedialer Navigationselemente, ist eine Verknüpfung von zeit- und strukturbasierter Navigation notwendig. Darüber hinaus muss bei der Konzeption einzelner Navigationselemente das gesamte

Studierendeninterface unter besonderer Berücksichtigung der Kompatibilität der einzelnen Navigationshilfen untereinander betrachtet werden.

### 4.5.3 Abgeleitete Anforderungen

Aus den in 4.5.1 und 4.5.2 erarbeiteten Anforderungen an die Anpassung hypermedialer Navigation für Vorlesungsaufzeichnungssysteme lassen sich eine Reihe von Rahmenbedingungen ableiten, die die Umsetzung eines hypermedialen Navigationskonzeptes ermöglichen. Im Folgenden werden diese Rahmenbedingungen dargestellt und es werden ggf. weitere Gründe, die für deren Realisierung sprechen, diskutiert.

#### 4.5.3.1 Skalierbarkeit verschiedener Elemente des Interfaces

Wie bereits in 4.5.1.2 angerissen, stellt die starke Verkleinerung von Übersichtsfolien in den meisten Systemen einen deutlichen Nachteil hinsichtlich deren Nutzbarkeit dar. Diesem Problem durch eine Vergrößerung der Übersichtsfolien zu begegnen, wirkt sich beim Abspielen der Aufzeichnung negativ auf die Darstellungsfläche der Hauptfolie oder des Videos aus. Als zweckmäßige Lösung erscheint daher die Möglichkeit zur flexiblen Skalierung einzelner Bestandteile des Studierendeninterfaces, wie in Abb. 4-8 dargestellt.

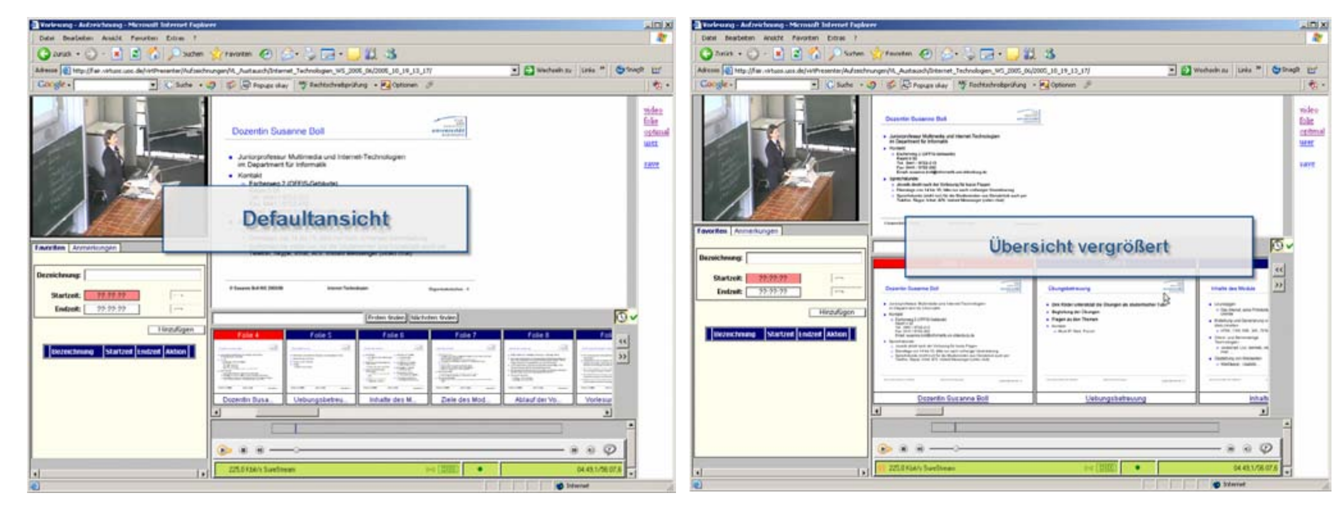



**Abb. 4-8: Flexible Skalierbarkeit einzelner Interfacekomponenten in einer Prototypversion des virtPresenter**

Ein weiterer Vorteil flexibler Skalierung einzelner Bestandteile des Interfaces ergibt sich aus der Tatsache, dass die Größe von Hauptfolie und Videofenster abhängig vom Vorlesungsgeschehen variiert werden kann.[37]

Ein wichtiger Punkt bei der Umsetzung der Skalierbarkeit einzelner Elemente des Interfaces ist die Berücksichtigung der Darstellungsqualität. Dies gilt in besonderem Maße für das Vergrößern von Folien [Hürst, Müller & Ottmann 2004]. Eine Möglichkeit, Folien in verschiedenen Zoomstufen mit gleich bleibend guter Qualität zu zeigen, ohne dabei inakzeptabel große Datenmengen zu verwenden, ist durch den Einsatz von Vektorgrafik gegeben.

[37] Im System DVDconnector wird versucht, einen ähnlichen Effekt durch zwei verschiedene Präsentationsmodi zu erreichen [Jantke 2005]. Dabei kann entweder die Hauptfolie oder das Videobild im Hauptfenster vergrößert gezeigt werden. Das System LODE, ein ePresence-Klon (vgl. 5.4), bietet drei verschiedene Größeneinstellungen für das verwendete Videofenster [Dolzani & Ronchetti 2005]. In beiden Ansätzen ist damit die Möglichkeit gegeben, das Videobild beispielsweise bei der Vorführung eines Experimentes zu vergrößern, bei der Suche in der Folienübersicht sind derartige Ansätze jedoch nicht hilfreich.

#### 4.5.3.2 Visualisierung zeitlicher Kontexte

Vor allem die in 4.5.1.3 geforderte zeitbasierte Realisierung von Footprints bedingt in Hinblick auf das Zusammenspiel der Navigationselemente untereinander die Notwendigkeit zur Einordnung sämtlicher Navigationshilfen und Links in zeitliche Kontexte. Auch die Betrachtung anderer Navigationshilfen in 4.5.1 spricht für eine Visualisierung des zeitlichen Kontextes, in den eine Navigationsaktion eingeordnet ist. Diese Forderung gilt vor allem für strukturbasierte Navigationsaktionen. So sollte beispielsweise bei der Anwahl eines strukturellen Elementes oder eines Folientitels in der Übersicht visualisiert werden, welcher Punkt auf der Zeitachse der zugehörigen Navigationsaktion zugeordnet werden kann. Darüber hinaus sollte die Tatsache berücksichtigt werden, dass z. B. Bookmarks oder auch strukturellen Elementen in vielen Fällen nicht nur ein Startpunkt sondern auch ein Endpunkt zugeordnet werden kann. Der Endpunkt eines strukturellen Elementes oder einer Folie lässt sich dabei durch den zeitlich folgenden Folienwechsel, bzw. bei strukturellen Elementen durch die nächste Folienanimation bestimmen.

#### 4.5.3.3 Visualisierung struktureller Kontexte

Alle primär zeitbasierten Navigationselemente müssen in strukturelle Kontexte eingebettet werden, um der Forderung nach der Möglichkeit zum Zusammenspiel der Navigationselemente nachzukommen (vgl. 4.5.2). So sollten beispielsweise Bookmarks die entsprechenden Abschnitte des Thumbnail-Überblicks zugeordnet werden können. Auch Footprints müssen direkt mit dem strukturellen Überblick verbunden werden, um zeitbasierte Markierungen bereits angesehener Abschnitte mit den zugehörigen Inhalten verbinden zu können. Die technische Realisierung eines solchen Vorhabens ist dank der Synchronisation von Audio oder Video mit der Folienpräsentation vergleichsweise einfach zu gestalten. Wichtig ist hierbei jedoch, dass strukturelle Informationen auch auf einer höher aufgelösten Ebene berücksichtigt werden sollten, was durch die Einbeziehung von Folienanimationen möglich ist (vgl. die einleitenden Abschnitte zu 4.5 und 4.5.1).

#### 4.5.3.4 Verbindung von Zeit und Struktur

Wie bereits größtenteils durch die vorangehenden beiden Punkte impliziert, ist zur Verwirklichung eines hypermedialen Navigationskonzeptes für Vorlesungsaufzeichnungen eine Verbindung von Zeit und Struktur essentiell. Zum einen ist dieser Sachverhalt durch die hybride Form von Vorlesungen und damit auch Vorlesungsaufzeichnungen selbst bedingt. In einer Vorlesung werden vorstrukturierte Inhalte zeitlich sequenziell hintereinander behandelt. Zum anderen stellt die Tatsache, dass einige Navigationshilfen primär zeitbasiert und andere primär strukturbasiert konzipiert sind, hinsichtlich der in 4.4.3 beschriebenen synergetischen Arbeitsweise einzelner hypermedialer Navigationselemente ein Hindernis dar, das nur durch die konsequente Verbindung von Zeit und Struktur im Navigationsinterface überwunden werden kann. In Anbetracht der in [Dufour, Toms, Lewis & Baecker 2005] geschilderten Evaluationsergebnisse, die gezeigt haben, dass eine unidirektionale Verbindung von Zeit zu Struktur nur zu mäßigem Erfolg führen kann, sollte die Verbindung von zeit- und strukturbasierter Navigation bidirektional realisiert werden, um die Vorteile beider Herangehensweisen kombinieren zu können (vgl. dazu auch 4.2.2.1).

#### 4.5.3.5 Animation von Folienschritten

Da anhand der Animation von Folienschritten auch die Feinstruktur eines Folienvortrags automatisch aufgezeichnet werden kann (vgl. 4.2.1.3), ist die korrekte Aufzeichnung und Wiedergabe der Zeitpunkte und der Reihenfolge von Folienanimationen nicht nur in Bezug auf die Implementation von strukturellen Elementen (vgl. 4.5.1.2) sondern auch hinsichtlich der in 4.5.3.3 beschriebenen Visualisierung struktureller Kontexte essentiell.

Ein weiterer Grund, der für die korrekte Abbildung von Folienanimationen spricht, ist die Tatsache, dass die Abbildung einer Sequenz von Animationen auf eine statische, nicht animierte Folie zum Verlust der Konsistenz von Wort und Bild führt und eine Reihe weiterer Nachteile mit sich bringt, die in der Einleitung 8.1 verdeutlicht werden. Auch entfielen bei einer derartigen Abbildung auf statische Folien die in 2.2.2 genannten Vorteile, die sich aus der Verwendung von Animationen in elektronischen Folienpräsentationen ergeben.

## 4.6 Fazit und Einordnung in den weiteren Verlauf

Im vorliegenden Kapitel wurde zunächst gezeigt, dass hypermediale Navigation ein viel versprechender Ansatz zur Navigation in Vorlesungsaufzeichnungen ist, der bisherigen Ansätzen in vielen Punkten überlegen ist. Vorteile dieses Ansatzes ergeben sich vor allem aus der Möglichkeit zur Nutzung struktureller Eigenschaften des verwendeten Folienmaterials (vgl. 4.3.1) und aus der Orientierungsunterstützung durch hypermediale Navigationselemente (vgl. 4.3.2). Im Anschluss an diese Betrachtung wurde eine Reihe von Anforderungen an die Anpassung hypermedialer Navigation für Vorlesungsaufzeichnungen und deren Implikationen für die Gestaltung eines geeigneten Interfaces herausgearbeitet.

Im weiteren Verlauf der Arbeit wird zunächst ein Überblick über bereits existierende Vorlesungsaufzeichnungssysteme gegeben, die einzelne hypermediale Navigationselemente wie beispielsweise Bookmarks oder Volltextsuche implementieren. Die Betrachtung der Systeme soll zum einen Anhaltspunkte für die Implementation einzelner hypermedialer Navigationshilfen im Kontext eines hypermedialen Navigationskonzeptes für Vorlesungsaufzeichnungen und zum anderen anhand der Verbreitung einzelner hypermedialer Navigationselemente den Bedarf eines umfassenden Navigationskonzeptes aufzeigen. Der Frage, warum ein hypermediales Navigationskonzept für Vorlesungsaufzeichnungen nicht bereits entwickelt wurde, wird in der Einleitung zu Kapitel 5 nachgegangen.

An den Überblick im nächsten Kapitel anschließend wird in Kapitel 1 aufbauend auf den im aktuellen Kapitel erarbeiteten Anforderungen ein hypermediales Navigationskonzept für Vorlesungsaufzeichnungen entwickelt.

.

# 5 Grundzüge hypermedialer Navigation in aktuellen Systemen zur Vorlesungsaufzeichnung – Ein Überblick über relevante Systeme

Seit Mitte der neunziger Jahre ist das WWW in seiner klassischen Form als Hypertext-Medium in immer mehr Arbeits- und Lebensbereichen zu finden [Leiner, Cerf, Clark, Kahn, Kleinrock, Lynch, Postel, Roberts & Wolff 2005; Griffiths 2002]. Ursprünglich als ein hypermediales Netzwerk zum Austausch und zur Verbreitung wissenschaftlicher Informationen konzipiert [Berners-Lee, Cailliau, Groff & Pollermann 1992], bietet es eine ideale Infrastruktur zum schnellen und vergleichsweise einfachen Zugriff auf Informationen jeglicher Art. Diese Tatsache wird von immer mehr Menschen in den verschiedensten Arbeits- und Lebensbereichen erkannt und genutzt. Medienkompetenz im Umgang mit dem Internet und daher mit hypermedial aufbereiteter Information kann aufgrund der weiten Verbreitung und des breiten Anwendungsspektrums des Mediums mittlerweile als eine Kulturtechnik betrachtet werden, deren Bedeutung der des Hörens/Sprechens, Lesens/ Schreibens und Rechnens nahe kommt [Zimmerli 2004].[38] So schreibt die Europäische Kommission der „digitalen Kompetenz“ im einundzwanzigsten Jahrhundert eine ähnliche Bedeutung zu wie den „klassischen Kompetenzen“ Rechnen, Schreiben und Lesen Anfang des zwanzigsten Jahrhunderts [Europäische Kommission 2002].

Angesichts dieser Alltäglichkeit des Arbeitens mit Hypermedien ist es kaum verwunderlich, dass Elemente hypermedialer Navigation auch in vielen bereits existierenden Systemen zur Aufzeichnung von Vorlesungsmitschnitten zu finden sind. Das vorliegende Kapitel gibt einen Überblick über Systeme mit weiterführenden Navigationsansätzen.[39] Bei der vergleichenden Betrachtung fällt auf, dass einzelne Elemente hypermedialer Navigation bereits in vielen Systemen zu finden sind. Ein umfassendes hypermediales Navigationskonzept wie in Kapitel 4 beschrieben ist jedoch in keinem der Systeme zu finden.

Diese Verbreitung einzelner und unterschiedlicher hypermedialer Navigationselemente zeigt nicht nur Möglichkeiten zur Implementation hypermedialer Navigation in Vorlesungsaufzeichnungssystemen sondern auch die Bedeutung hypermedialer Navigation in Vorlesungsaufzeichnungen. Der vorliegende Abschnitt stellt damit nicht nur einen Überblick über die derzeit hinsichtlich Navigation wichtigsten Vorlesungsaufzeichnungssysteme und Ansätze dar. Er soll darüber hinaus zur Illustration der Relevanz hypermedialer Navigation für Vorlesungsaufzeichnungen dienen. Besonders aussagekräftig ist dabei die Beobachtung, dass alle derzeit bedeutenden Systeme Elemente hypermedialer Navigation aufweisen. Die Frage, warum im Rahmen der Entwicklung der vorgestellten Systeme kein konsistentes hypermediales Navigationskonzept für Vorlesungsaufzeichnungen entwickelt wurde, lässt sich vor allem dadurch beantworten, dass es sich bei vielen der implementierten Navigationsansätze eher um Ad-hoc-Entwicklungen handelt, die aus der Notwendigkeit heraus entstanden sind (vgl. beispielsweise 5.6). Navigation als Forschungsfrage ist erst vergleichsweise spät in den Fokus vieler Entwickler geraten, da im Forschungsfeld Vorlesungsaufzeichnungen zunächst eher technische Fragestellungen im Vordergrund standen [Hürst, Müller & Ottmann 2004].

---

[38] Der Philosoph und Gründungspräsident der Volkswagen AutoUni Walther Zimmerli hat diesen Gedanken bereits Ende des letzten Jahrhunderts im Bezug auf rechnerverwaltete Information formuliert. In [Zimmerli 2004] konkretisiert er seine Aussage dahingehend, dass der Umgang mit dem Internet den Status einer neuen Kulturtechnik einnimmt. Letztendlich ist das Internet in seiner heutigen Form in erster Linie nicht mehr als hypermediales Informationsnetzwerk. Daher kann also nicht nur der Umgang mit vernetzten Rechnern, sondern der Umgang mit hypermedial aufbereiteter Information als Kulturtechnik verstanden werden.

[39] Über die in diesem Teil der Arbeit beschriebenen Systeme hinaus existieren eine Vielzahl von weiteren Systemen zur Vorlesungsaufzeichnung, die jedoch nur über rudimentäre Navigationsmöglichkeiten verfügen.

Eine weitere Erklärung für den Mangel an konsistenten hypermedialen Navigationskonzepten im Bereich Vorlesungsaufzeichnungen mag in der bereits weiter oben erwähnten Tatsache liegen, dass der Umgang mit Hypermedia mittlerweile als Kulturtechnik betrachtet werden kann und als solche ein gewisses Maß an Transparenz erreicht hat. Kulturtechniken wie Sprechen/Hören, Lesen/Schreiben und Rechnen werden – von Ausnahmen, wie beispielsweise der psychologischen oder linguistischen Forschung abgesehen – in der Regel nicht reflektiert, sondern einfach benutzt. Paradoxerweise werden Kulturtechniken immer weniger analysiert, je mehr sie genutzt werden. Unterbewusst mögen daher viele der Entwickler der im Verlauf dieses Kapitels betrachteten Systeme versucht haben, das eine oder andere hypermediale Navigationselement umzusetzen. Eine konzeptionell konsequente Umsetzung blieb jedoch aus, da nicht versucht wurde, hypermediale Navigationskonzepte für Vorlesungsaufzeichnungen zu entwickeln, sondern nur Teilaspekte übernommen wurden. Interessanterweise ist in den jeweiligen Veröffentlichungen zu den vorgestellten, im Rahmen dieser Arbeit als hypermedial klassifizierten, Navigationsmechanismen auch kein Verweis auf äquivalente Hilfen (von Volltextsuche abgesehen) in anderen Medien wie beispielsweise Hypertext zu finden. Vielmehr werden die jeweiligen Navigationselemente von den Autoren als Neuentwicklungen verstanden. Auch diese Tatsache kann als Indiz für die Transparenz hypermedialer Navigation gewertet werden.[40]
Zur folgenden Betrachtung der einzelnen Systeme wird neben dem Studierendeninterface und damit der Navigationsfunktionalität jedes der Systeme auch der Produktionsprozess einer Aufzeichnung mit dem jeweiligen System eingehender beschrieben. Die Motivation dieser Herangehensweise liegt vor allem in der Tatsache, dass viele Navigationsansätze mit manueller Indizierung der Aufzeichnung oder sonstiger manueller Nachbearbeitung einhergehen.

## 5.1 MANIC

Hinter dem Akronym MANIC verbirgt sich *Multimedia Asynchronous Networked Individualized Courseware*, ein vergleichsweise frühes Konzept der Universität von Massachusetts in Amherst zur Aufzeichnung von Vorlesungen. MANIC stellt den Studierenden RealAudio Dateien mit synchronisierten HTML-Seiten, bzw. statischen GIF's zur Verfügung [Stern, Steinberg, Lee, Padhye & Kurose 1997].[41] Zur grundlegenden Philosophie bei der Entwicklung des Systems gehört es, auf Client-Seite keine spezielle Software, sondern nur frei verfügbare Browser-Plug-Ins, wie beispielsweise RealAudio zu verwenden [Stern, Steinberg, Lee, Padhye & Kurose 1997]. Das System selbst ist Open Source Software [Schapira, De Vries & Pedregal-Martin 2001].
Besondere Beachtung verdient MANIC aufgrund der beiden Tatsachen, dass die Wiedergabe-Komponente des Systems mehrere gut durchdachte Navigationseigenschaften integriert [Stern, Steinberg, Lee, Padhye & Kurose 1997] und dadurch, dass das System in einem größeren Verbund von Universitäten eingesetzt wird [Adrion, Burleson, Cooper, Israel, Kurose & Watts 2003].

[40] Bei der Anpassung von elastischen Interfaces, einem eher unbekannten Ansatz zum Scrollen elektronischer Textdokumente an AOF (siehe 5.2), sind sich die Entwickler beispielsweise sehr genau der Tatsache bewusst gewesen, dass sie einen vorhandenen Ansatz anpassen [Hürst, Götz & Janvers 2004].
[41] Neben dieser Web-Ausgabe des Systems existiert auch eine CD-Variante, die Video-Aufnahmen ermöglicht [De Vries & Thampuran 2001; Adrion, Burleson, Cooper, Israel, Kurose & Watts 2003]. Die Grundstruktur der Ausgabe (HTML und Audio-Dateien) bleibt dabei unverändert. In [Schapira, De Vries & Pedregal-Martin 2001] wird auch das Streaming von Video erwähnt. Ein Blick auf die Homepage des Projektes lässt jedoch erkennen, dass die Verwendung von Streaming-Video bei MANIC nach anfänglichen Versuchen eingestellt wurde.

### 5.1.1 Produktionsprozess

Die während der Lehrveranstaltung verwendeten Materialien müssen vor der Veranstaltung in einem speziellen Autorenwerkzeug generiert werden. Das Autorenwerkzeug ist web-basiert und aufgrund seiner einfachen Struktur sehr leicht zu bedienen [De Vries & Thampuran 2001]. Der Nachteil dieses Verfahrens liegt darin, dass evtl. verwendete Grafiken nicht innerhalb des Tools erstellt und verändert werden können. Änderungen verlangen daher externe Bearbeitung und erneutes Hochladen der jeweiligen Grafik. Auch Animationen, wie beispielsweise der schrittweise Aufbau einer Grafik sind damit nicht möglich.

Abb. 5-1 zeigt einen Screenshot der Eingabemaske des Autorentools. Minimale Formatierungsmöglichkeiten wie Schriftgröße und -farbe sind im System gegeben. Erweiterte Formatierung ist durch die direkte Eingabe von HTML-Code zu erreichen. Dadurch lassen sich beliebige HTML-Elemente, wie beispielsweise Java-Applets in die Folien integrieren [De Vries & Thampuran 2001]. Es hat sich jedoch gezeigt, dass eine Vielzahl der Benutzer nicht über die Fähigkeit verfügt, auch nur simple Eingaben direkt in HTML vorzunehmen und daher auf die Eingabemaske angewiesen ist [De Vries & Thampuran 2001].




**Abb. 5-1 Eingabemaske des MANIC Autorentools**[42]

Ein Nachteil dieser Technik ist, dass HTML-ungeübte Dozenten im Vergleich zur PowerPoint-basierten Erstellung der Vorlesungsmaterialien stark eingeschränkt werden. Ihnen stehen bei der Aufbereitung des Lehrstoffs nur wenige Freiheitsgrade zur Verfügung. Ein weiteres Problem sind Sonderzeichen und mathematische Formeln, die entweder als Bilder oder MathML eingebunden werden müssen.[43] Zur Erstellung von Bildern sind dabei

[42] Aus Gründen der Lesbarkeit wurde die Farbgebung nachträglich verändert.

[43] MathML ist zum heutigen Zeitpunkt noch nicht besonders weit verbreitet. Die Darstellbarkeit beschränkt sich auf wenige Browser.

externe Werkzeuge nötig. Auch zur Eingabe von MathML wird aufgrund der komplexen Eingabe von Formeln die Verwendung von speziellen Editoren empfohlen [44].
Gerade in Fällen, in denen Dozenten viel Wert auf Details legen, kann dieser mehrstufige Editionsprozess von Formeln und Grafiken erheblich mehr Zeit in Anspruch nehmen als die Erstellung der Materialien in geeigneter Präsentationssoftware wie beispielsweise PowerPoint.
Andererseits bietet das System jedoch eine relativ einfache Plattform zur Erstellung multimedialer Lerninhalte, gerade auch für technisch weniger versierte Dozenten. In [Dickson, Adrion & Hanson 2006] wird eine alternative Aufnahmekomponente des Systems vorgestellt, mit der beliebige Bildschirminhalte aufgezeichnet und indiziert werden können.

### 5.1.2 Studenteninterface

Da es sich bei MANIC um HTML-Seiten mit gekoppelten RealAudio-Dateien handelt [Stern, Steinberg, Lee, Padhye & Kurose 1997], verfügt das System zumindest im Ansatz über alle hypermedialen Navigationseigenschaften, die im jeweiligen Browser implementiert sind. So können beispielsweise Bookmarks auf einzelne Folien gesetzt werden.[44] Auch die Verwendung des browser-eigenen Backtracking ist möglich.
Diese Navigationseigenschaften beziehen sich jedoch bei näherem Hinsehen nur auf die HTML-Seiten einer Aufzeichnung. Die Kopplung von RealAudio an eine HTML-Seite lässt bei Anwahl der Seite über Bookmarks oder Backtracking die jeweilige Audio-Datei abspielen. Da der Browser jedoch nur Informationen über die HTML-Seite und nicht über die Abspielposition des RealPlayers beim Abspielen der Audiodatei speichert, wird die Audio-Datei von der Anfangsposition her abgespielt. Eine genaue Zuordnung von Bookmarks oder Backtracking-Aktionen zu einem Zeitindex innerhalb der Aufzeichnung ist damit nicht möglich. Auch die in 4.3.4.2 beschriebe Problematik der Zuordnung von Footprints zu einzelnen Passagen des Mitschnitts bleibt bestehen.

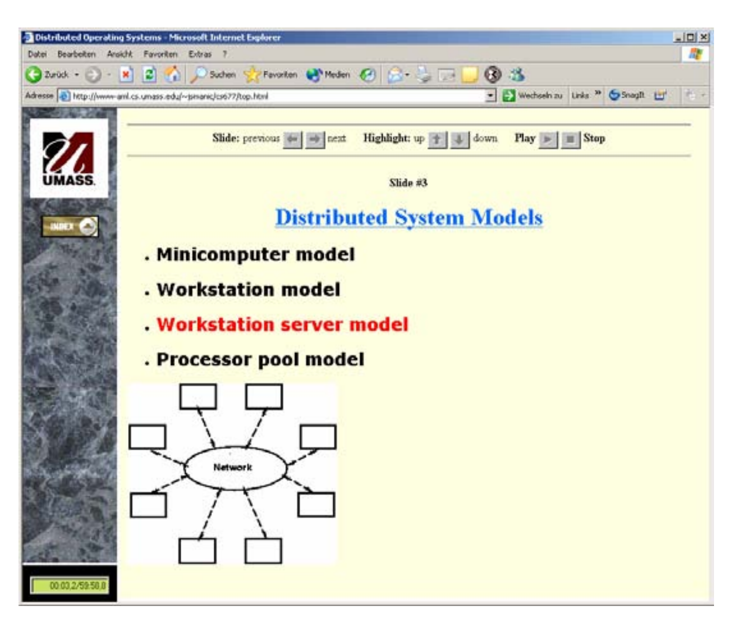



**Abb. 5-2: Screenshot des MANIC Studenteninterfaces**

[44] Die Möglichkeit, Bookmarks oder Links auf einzelne Seiten zu setzen, wird im System nicht direkt angeboten. Jeder animierte Bestandteil wird jedoch über eine eigene Adresse angesteuert. Durch Öffnen der Folie in einem separaten Fenster lässt sich die Adresse anzeigen. Durch Paste and Copy der referenzierten Adresse ist es findigen Benutzern so möglich, Links und Bookmarks zu verwenden.

Neben diesen eher zufällig vorhandenen hypermedialen Navigationsfunktionen bietet MANIC auch eine Reihe von explizit eingearbeiteten Navigationsstrukturen und Orientierungshilfen. So gibt es in MANIC die Möglichkeit, jeweils zum vorherigen und zum nächsten thematischen Punkt der Folie und zu den beiden direkt benachbarten Folien zu springen [Stern, Steinberg, Lee, Padhye & Kurose 1997]. Auch wird, wie in Abb. 5-2 gezeigt, das jeweils aktuelle Element einer Bullet-Liste farblich hervorgehoben. Dieses Highlighting dient zur gezielten Fokussierung der Aufmerksamkeit [Stern, Steinberg, Lee, Padhye & Kurose 1997] und erfüllt damit eine wichtige didaktische Funktion.

In neueren Versionen verfügt MANIC auch über eine Volltext-Suche [Schapira, De Vries & Pedregal-Martin 2001]. Obwohl MANIC prinzipiell eine Verlinkung der Suchergebnisse auf die jeweiligen farblich hervorgehobenen Bestandteile einer Folie erlauben würde, ist die Verlinkung nur foliengenau implementiert. Die Feinschritte der Navigation bleiben also dem Benutzer überlassen. Für die Navigation auf höherer Ebene verfügt das System über ein verlinktes Inhaltsverzeichnis.
Durch eine *previous item – next item*-Navigation bietet MANIC die Möglichkeit, jeweils zum vorherigen und zum nächsten thematischen Punkt der Folie und zu den beiden direkt benachbarten Folien zu springen.
Ein Überblick über alle in MANIC implementierten Navigationseigenschaften wird in Tabelle 5-1 gegeben.

<table>
<tr><td rowspan="8">Hypermediale Navigation</td><td rowspan="5">Allgemein</td><td>Backtracking</td><td>✓</td><td rowspan="3">Jeweils nur zum Anfang eines Animationsschrittes</td></tr>
<tr><td>Bookmarks</td><td>✓</td></tr>
<tr><td>Links</td><td>✓</td></tr>
<tr><td>Footprints</td><td>-</td><td></td></tr>
<tr><td>Volltextsuche</td><td>✓</td><td>Direktes Ansteuern des gefundenen Elementes ist nicht möglich</td></tr>
<tr><td rowspan="3">Strukturelle Eigenschaften</td><td>Inhaltsverzeichnis</td><td>✓</td><td>Ist Inhaltsverzeichnis wirklich eine strukturelle Eigenschaft?</td></tr>
<tr><td>„prev“-„next“-Navigation</td><td>✓</td><td></td></tr>
<tr><td>Folienbasierte Navigation</td><td>✓</td><td></td></tr>
<tr><td colspan="3">Zeitbasierte Navigation</td><td>-</td><td></td></tr>
</table>

**Tabelle 5-1: Navigationseigenschaften des MANIC-Systems**

### 5.1.3 Fazit und weitere Informationen

MANIC ist Open Source, verfügt über ein eigenes Autoren-Tool und implementiert, wenn auch eher zufällig, eine Reihe von hypermedialen Navigationseigenschaften. Das Navigationskonzept des Systems ist prinzipiell gut durchdacht, es fehlt jedoch die Möglichkeit zur zeitbasierten Navigation. Volltextsuche ist möglich, animierte Bestandteile sind damit jedoch nicht direkt ansteuerbar. Die gegebenen Elemente zur hypermedialen Navigation sind nur mit viel Einfallsreichtum anzusteuern und beziehen sich jeweils nur auf den Anfang eines animierten Folienbestandteils. Durch das Fehlen von Footprints fallen, wie bereits in 4.5.2 erläutert, entscheidende Vorteile hypermedialer Navigation weg.
Auf der Homepage des Projekts [16] sind eine Vielzahl von Informationen und im Rahmen der Entwicklung von MANIC entstandenen Publikationen sowie zahlreiche Beispielkurse zu finden. Darüber hinaus bietet die Seite die Möglichkeit, in MANIC eigene Beispielkurse anzulegen.

## 5.2 AOF

Die Abkürzung AOF steht für *Authoring on the Fly*, den Grundgedanken jeder Vorlesungsaufzeichnung. Das Konzept wurde an der Albert-Ludwigs-Universität Freiburg entwickelt und zusammen mit einer ersten Implementation bereits 1995 in [Bacher & Ottmann 1995] der wissenschaftlichen Öffentlichkeit vorgestellt. Anfangs basierte das System auf der Aufzeichnung von Audio bzw. Video, PostScript-Folien und handschriftlichen Annotationen [Bacher & Ottmann 1995].[45] Mittlerweile können auch Folien anderer Standardgrafikformate in das System importiert werden [Hürst, Maass, Müller & Ottmann 2001]. Die Verwendung von PowerPointfolien wird durch das Konvertierungsprogramm *ppt2aof* ermöglicht. Bei der Konvertierung werden die Grundeigenschaften der Folienpräsentation in das AOF-Format übernommen [Lauer & Ottmann 2002].
Neben der Aufzeichnungskomponente sind im Projekt eine Reihe von zusätzlichen Komponenten wie eine speziell auf AOF-Dokumente abgestimmte Suchmaschine [Hürst 2002] und ein mobiles Aufzeichnungs- und Vortragspult [Belenkaia, Lienhard, Mohamed & Ottmann 2004] entwickelt worden. Darüber hinaus wurde auch an der Entwicklung verschiedener Interfaces für Folienpräsentationen gearbeitet [Hürst & Meyer 2004]. Mit *aofedit* verfügt das System über eine Erweiterung, die es erlaubt, bereits erstellte Aufzeichnungen nachträglich unter Beibehaltung der erzeugten Synchronisationsinformationen von Tonspur und Folien zu schneiden [AOFedit 2004]. Mit *AOF2Flash* existiert sogar eine Möglichkeit zur Konvertierung von AOF-Folien in Flash [Rechert 2003].[46,47] Dieser Ansatz verbessert jedoch lediglich die Darstellungsqualität der angezeigten Folien. Skalierbarkeit der Navigationsansicht, Volltextsuche oder Möglichkeiten zur interaktiven Manipulation der in der Navigationsansicht gezeigten Folien sind in dem mit *AOF2Flash* erzeugten Interface nicht gegeben.
Dem Problem der Navigation in Vorlesungsaufzeichnungen wird in AOF mit einem eigenen Ansatz, dem so genannten *random visible scrolling* begegnet [Bacher & Müller 1998], die im Abschnitt über das Studenteninterfaces von AOF eingehender betrachtet wird.
AOF wurde in den universitätsübergreifenden Projekten Viror (virtuelle Hochschule Oberrhein) [43] und ULI (universitärer Lehrverbund Informatik) [42] eingesetzt. In Viror befand sich das System allerdings noch in der Testsphase und auch in ULI wurde es aufgrund unterschiedlicher Anforderungen der Projektpartner nicht an allen Standorten genutzt [Kandzia, Kraus & Ottmann 2004].
Mit Lecturnity sind die Erfahrungen des AOF- Projektes in die Entwicklung eines kommerziellen Produktes eingeflossen [Lauer & Ottmann 02].

### 5.2.1 Produktionsprozess

Zur Aufzeichnung eines Vortrages stehen die beiden Programme *AOFwb* (AOF-Whiteboard) und *mlb* (Multimedia Lecture Board) zur Verfügung. Das *mlb* ist eine Weiterentwicklung des AOFwb und verfügt zusätzlich zur Aufnahmefunktion über Möglichkeiten zur Telekooperation [Effelsberg, Walke, Stüttgen & Ottmann 2003]. Im Gegensatz zum *AOFwb*, das nur in einer Linux-Version existiert, ist das *mlb* auch in einer unter Windows lauffähigen Version implementiert. Zur Verwendung von PowerPoint-Folien bietet sich daher auf den ersten Blick am ehesten die Verwendung des Konvertierungsprogramms *ppt2mlb* und die anschließende Aufzeichnung im *mlb* an.

---

[45] Handschriftliche Annotationen dienen dabei vor allem zur Verdeutlichung des in der Vorlesung Gesagten. PowerPoint-Animationen haben in neueren Systemen den selben Effekt.
[46] Die in [Rechert 2003] vorgestellten Ergebnisse scheinen auf [Kaindl 2002] aufzubauen.
[47] Auch SVG wurde für den Einsatz in AOF evaluiert. Das Format wurde verworfen, da die Entwickler die komplette Präsentation in einer Datei speichern wollten und da dieses Vorgehen aufgrund der resultierenden Dateigrößen zu Problemen bei der Darstellung führt [Kaindl 2002].

Da die aktuell verfügbaren Versionen sowohl von *ppt2mlb* als auch von *ppt2aof* jedoch nur in PowerPoint 2000 lauffähig sind, gibt es für Benutzer neuerer PowerPoint-Versionen keinen Weg zur Konvertierung der bereits erstellten Folien.
Auch in der Konvertierung von Folien, die mit PowerPoint 2000 erstellt wurden, weist *ppt2mlb* in der aktuell verfügbaren Version 0.85 Schwachstellen auf. So werden Animationen nicht umgewandelt, die Folie erscheint im *mlb* also nur in der Druckansicht. Dies kann wie in der Einleitung zu 8.1 dargestellt zu unbeabsichtigten Überdeckungen von Folienbestandteilen und zu Störungen des Aufmerksamkeitsflusses der Zuhörer führen. Die Vorteile der Präsentationssoftware PowerPoint als Medium sind im *mlb* damit nicht mehr vorhanden. Auch Probleme mit der Umsetzung halbtransparenter Objekte führen zu ungewünschten Effekten, wie in Abb. 5-3 zu sehen. Der Vergleich mit einem Export derselben Folie aus PowerPoint 2000 in das GIF-Format zeigt, dass es sich hierbei nicht um einen Fehler der Exportfunktion von PowerPoint 2000 handelt.




**Abb. 5-3: Screenshot des *mlb* und Export derselben Folie aus PowerPoint 2000 als GIF im Vergleich**[48]

Nach der Konvertierung können die Folien im *mlb* präsentiert werden. In der Version 1.0.1 gibt es einen Full-Screen-Modus, der die Präsentation wie eine PowerPoint-Präsentation aussehen lässt. Bewegungen des Mauszeigers, während des Vortrags angefertigte Zeichnungen und bei Vorhandensein entsprechender Eingabegeräte handschriftliche Annotationen werden zusammen mit dem aufgenommenen Ton und den Zeitinformationen der Folienwechsel aufgezeichnet. Allerdings ist anzumerken, dass die Verwendung von grafikintensiven Folien mit Hintergrundbild bei Tests wiederholt zum Absturz der Software während der Aufnahme geführt hat.
Der gangbare, wenn auch etwas umständlichere Weg ist daher die Verwendung des *ppt2aof* Konvertierungsprogramms und die anschließende Aufzeichnung des Vortrags im *AOFwb*.
Der Export von Animationen funktioniert hier,[49] wenn auch das Handling der Animationen im *AOFwb* etwas gewöhnungsbedürftig ist. Das *AOFwb* funktioniert zuverlässig und zeichnet wie auch das *mlb* Bewegungen des Mauszeigers und während des Vortrags gemachte Zeichnungen bzw. handschriftliche Annotationen auf. Im *AOFwb* ist im Gegensatz zum *mlb* kein Full-Screen-Modus enthalten, was zu einer etwas ungewöhnlichen Art der Präsentation führt. Auch der Einsatz unterschiedlicher Betriebssysteme für Erstellung und Verwendung

[48] Die unterschiedlichen Seitenzahlen auf den Folien sind kein Artefakt, sondern sind aufgrund eines dokumentierten Bugs in der Konvertierungssoftware entstanden.
[49] Bei den exportierten Animationen handelt es sich jedoch nur um die in PowerPoint 2000 verfügbaren Animationen. Selbst bei Vorhandensein einer PowerPoint XP und einer PowerPoint 2000 Version gehen Effekte wie das Verschwinden von Objekten, Hervorhebungen und Animationspfade beim Öffnen der mit PowerPoint XP erstellten Präsentation mit PowerPoint 2000 verloren.

der Vortragsmaterialien kann problematisch sein. Der in Abb. 5-3 gezeigte Darstellungsfehler halbtransparenter Objekte tritt beim *AOFwb* genau so auf, wie beim *mlb*.[50]
In der Standardeinstellung werden Video-Aufnahmen des Vortrags vom *AOFwb* nicht unterstützt, sie können jedoch als externe Medien eingebunden werden [AOFwb 2000].
Die Nachbearbeitung der mit *AOFwb* oder *mlb* entstandenen Aufzeichnungen ist mittels *aofedit* relativ einfach. Die Software erlaubt Schnitte innerhalb der Aufzeichnung und das nachträgliche Verschieben von Synchronisationsevents. Bei Verwendung von Copy und Paste gehen allerdings Synchronisationsinformationen verloren [AOFedit 2004].

### 5.2.2 Studenteninterface

In [Lauer & Ottmann 02] wird die Möglichkeit zur Konvertierung von AOF-Aufzeichnungen in Real als internettaugliches Wiedergabeformat erwähnt. Dabei gehen allerdings Annotationen und auch die Möglichkeit zum *random visible scrolling* verloren [Lauer & Ottmann 02]. Auch erfolgt die Umwandlung nicht automatisch [Lauer & Ottmann 02]. Da zudem keine Dokumentation dieser Umwandlung in den entsprechenden Handbüchern [AOFwb 2000; AOFedit 2004] zu finden ist[51], werden im Folgenden nur die AOF-Betrachterprogramme *aofSync* und *aofJSync* behandelt. Ein leicht abgewandeltes Benutzerinterface für die Verwendung in PDAs wird in [Zupancic 2006] beschrieben.

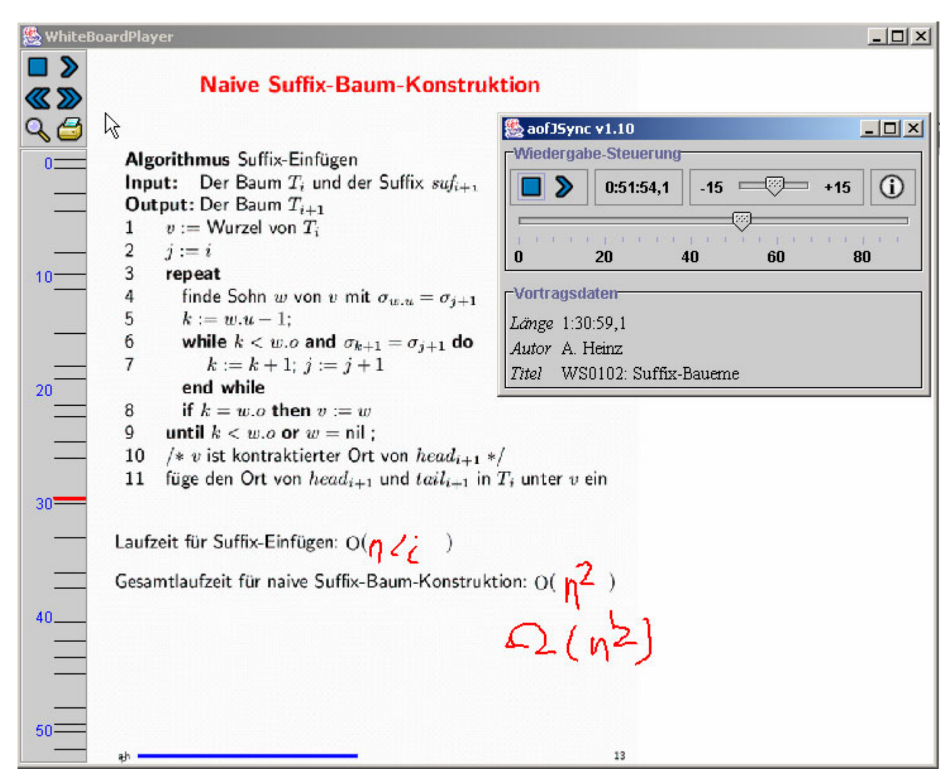



**Abb. 5-4: Screenshot des Studenteninterfaces *aofJSync*[52]**

Die beiden Programme sind hinsichtlich ihrer Benutzerinterfaces weitestgehend identisch, *aofSync* existiert jedoch in mehreren plattformspezifischen Varianten, wohingegen es sich bei *aofJSync* um eine in Java implementierte Weiterentwicklung handelt. Wie in Abb. 5-4 zu

[50] Ein Blick in die zugrunde liegende Speicherformate zeigt, dass sowohl im *mlb* als auch im *AOFwb* die jeweiligen PowerPointfolien nicht als eine Grafik importiert werden, sondern aus einzelnen Teilgrafiken aufgebaut werden. Der Fehler scheint also bei der Rekonstruktion der Folie zu liegen.
[51] Auch die unter [2] zum Download bereit gestellte Software *aof2html* dient nicht zur Umwandlung in ein internetgerechtes Format, sondern zur Erstellung einer HTML-Übersichtsseite über mehrere AOF-Dokumente.
[52] Das im Screenshot gezeigte Material kann unter [1] herunter geladen werden.

sehen ist, besteht der Player aus zwei verschiedenen Fenstern. Das kleinere Fenster dient dabei der Lautstärkeregelung und der zeitbasierten Navigation mittels eines Sliders, das größere der Darstellung und der strukturelle Navigation mittels zweier *prev/next*-Item-Bedienelemente in der linken oberen Ecke und einer Zeitleiste am linken Rand des Fensters.
In der Zeitleiste sind Events wie Folienwechsel oder das Erscheinen animierter Elemente markiert. Beliebige Punkte innerhalb der Zeitleiste können durch Anklicken direkt angesteuert werden. Dies entspricht dem Anklicken von Folien, bzw. Folientiteln in anderen Systemen, erlaubt jedoch zusätzlich das Ansteuern von Zwischenpunkten. Die jeweils aktuelle Position wird durch einen roten, wagerechten Balken markiert. Der Slider im Navigationsfenster ist an die von den AOF-Entwicklern konzipierte Navigationshilfe des *random visible scrolling* gekoppelt. Das bedeutet, dass jede Positionsänderung am Slider die sofortige Aktualisierung der Foliendarstellung bewirkt. Dies bezieht sich auch auf die Position des Mauszeigers oder eventuell aufgezeichnete handschriftliche Annotationen. Da sowohl Bewegungen des Mauszeigers als auch handschriftliche Annotationen nicht als Momentaufnahme sondern animiert wiedergegeben werden, ist es mit dem Slider so möglich, handschriftliche Annotationen beschleunigt aufbauen zu lassen oder auch schrittweise zurückzufahren. Auch entlang des vom Mauszeiger zurückgelegten Pfades kann so relativ frei navigiert werden. Ein Nachteil dieser recht komfortablen Unterstützung von handschriftlichen Annotationen wird bei Betrachtung der Annotationen im unteren Bilddrittel von Abb. 5-4 deutlich. Dozenten setzen handschriftliche Annotationen auch in Situationen ein, die bereits bei der Erstellung der Folien vorhersehbar sind. In derartigen Situationen erfüllt eine Folienanimation denselben Zweck vielleicht sogar besser. Die Verwendung von Annotationen in einem solchen Fall vermindert, gerade bei Fehlen der entsprechenden Eingabegeräte, unbegründet die Lesbarkeit entscheidender Folienbestandteile.
Eine aufgrund der Zeitleise redundante Navigationshilfe stellt der blaue Balken am unteren Rand des Fensters dar. Er zeigt immer den Abstand vom Anfang der Aufzeichnung bis zur momentan gespielten Position an und bietet somit eine zusätzliche zeitliche Orientierung. Es handelt sich hierbei also nicht um eine Navigationshilfe, die mit der in 6.2.1 dargestellten, um Footprints erweiterten Zeitleiste verwandt ist.

| | | | | |
|---|---|---|---|---|
| **Hypermediale Navigation** | **Allgemein** | **Backtracking** | - | |
| | | **Bookmarks** | - | |
| | | **Links** | - | |
| | | **Footprints** | - | |
| | | **Volltextsuche** | - | **Es existiert jedoch eine externe Suchmaschine** |
| | **Strukturelle Eigenschaften** | **Inhaltsverzeichnis** | - | |
| | | **„prev“-„next“-Navigation** | ✓ | |
| | | **Folienbasierte Navigation** | ✓ | **Über die klickbare Leiste…** |
| **Zeitbasierte Navigation** | | | ✓ | |
| **Besonderheiten** | | | ***Random visible scrolling*** | |

**Tabelle 5-2: Navigationseigenschaften des AOF-Viewers *aofJSync***

Neben den genannten Navigationshilfen sind in der derzeit verfügbaren Version keine weiteren vorhanden, insbesondere auch keine im Player integrierte Volltextsuche. Volltextsuche

über mehrere AOF-Dokumente ist jedoch mit der AOF-Suchmaschine *aofSE* möglich [Hürst 2002].[53] Tabelle 5-2 fasst die Navigationseigenschaften des AOF-Viewers zusammen.
Im Rahmen des AOF-Projektes sucht die Multimedia-Forschungsgruppe der Universität Freiburg darüber hinaus auch nach weiteren Navigationsmöglichkeiten, die auf dem Prinzip des *random visible scrolling* basieren. So wurde eine Reihe von Interfaces entwickelt, in denen die zeitliche Auflösung der Zeitachse verändert werden kann, um eine feinkörnigere Navigation zu ermöglichen [Hürst & Stiegeler 2002]. Getestet wurden dabei nichtlineare Skalierungen sowie verschiedene Möglichkeiten zur manuellen und automatischen Einstellung der Navigationsgeschwindigkeit. Da diese Interfaces allerdings weniger intuitiv zu bedienen sind als herkömmliche zeitbasierte Navigationsmechanismen [Hürst & Stiegeler 2002], ist im weiteren Verlauf der Ansatz des *elastic panning* entwickelt worden [Hürst & Götz 2004]. Es handelt sich dabei um eine Anpassung so genannter elastischer Interfaces, die zunächst als Navigationshilfe für größere Datenmengen in statische Medien konzipiert worden sind [Masui, Kashiwagi & Borden 1995]. Dem Grundgedanken elastischer Interfaces folgend, bietet *elastic panning* beim *random visible scrolling* die Möglichkeit links und rechts neben den Navigationsslider zu klicken. Klicken Benutzer direkt neben den Slider, bewegt sich der Slider sehr langsam und damit mit sehr feinkörniger zeitlicher Auflösung in Richtung der Stelle, an die geklickt wurde. Wird der Mauszeiger bei gedrückter rechter Maustaste vom Slider entfernt, bewegt sich der Slider schneller und damit mit grobkörnigerer zeitlicher Auflösung. Wird der Mauszeiger an eine Position links vom Slider bewegt, scrollt das Interface rückwärts.
Im Wesentlichen ist *elastic panning* durch diese variable Auflösung des Scrollverhaltens eine Technik, die es erlaubt *random visible scrolling* auch in größeren Aufzeichnungen zur feinkörnigen Navigation einzusetzen. Durch den Zusammenhang zwischen Navigation anhand der Dokumentstruktur und zeitbasierter Navigation mit *random visible scrolling* ist damit letztendlich, wenn auch nicht in vollem Umfang, eine feinkörnige strukturelle Navigation möglich. Im derzeit verfügbaren AOF-Viewer ist *elastic panning* nicht implementiert. Aufgrund des reinen Zusatzcharakters und der Tatsache, dass *elastic panning* keine zusätzlichen Anforderungen an die Struktur der Aufzeichnung stellt, scheint eine Integration dieser Technik in zukünftige Versionen der AOF-Viewers oder des Lecturnity-Players jedoch wahrscheinlich.

### 5.2.3 Fazit und weitere Informationen

Im Rahmen von AOF sind an der Albert-Ludwigs-Universität Freiburg bemerkenswerte Ergebnisse im Kontext von Vorlesungsaufzeichnungen entstanden. In die Entwicklung des Systems sind viele Erfahrungen eingeflossen, die im jahrelangen Einsatz von AOF, vor allem an der Hochschule, an der es entwickelt wurde, gemacht worden sind. Wohl auch aufgrund der Tatsache, dass mit Lecturnity quasi eine kommerzielle Variante des Systems existiert, scheint AOF in erster Linie ein Forschungsprojekt zu sein.
Mit *random visible scrolling* und *elastic panning* zeigen die Entwickler von AOF Wege auf, zeitbasierte Navigation und in geringerem Maße auch strukturbasierte Navigation innerhalb von Vorlesungsaufzeichnungen erheblich zu verbessern. Da diese beiden Ansätze konzeptuell orthogonal zu dem im Rahmen dieser Arbeit vorgestellten hypermedialen Navigationskonzept für Vorlesungsaufzeichnungen sind, ist der kombinierte Einsatz aller drei Ansätze in einem Interface ohne weiteres machbar. Dies wird unter anderem auch durch die Einbettung von *random visible scrolling* in das in Kapitel 1 vorgestellte Benutzer-Interface deutlich.

[53] In einem Prototyp der Suchmaschine ist sogar Volltextsuche in der Audioaufzeichnung implementiert [Hürst 2003a], die anscheinend foliengenau, jedoch nicht wortgenau in die Aufzeichnung verlinkt. Zur Verbesserung der Spracherkennung nutz die Suchmaschine vorhandenen Folientext.

Die genaue Betrachtung der Aufzeichnungssoftware und der im AOF-Viewer implementierten Suchparadigmen zeigt, dass die Entwicklung von AOF sehr annotationszentriert verlaufen ist. Im Gegensatz zur in dieser Arbeit vorliegenden dokumentstruktur-orientierten Entwicklung wurde dabei sehr viel Wert auf die Aufzeichnung und Wiedergabe von Annotationen gelegt. Die Feinstruktur, die sich durch Animation einzelner Folienelemente ausdrückt, wurde nur in sehr geringem Umfang berücksichtigt.
Die Tatsache, dass AOF einen Großteil der Funktionalität von PowerPoint neu implementiert, erscheint auf den ersten Blick nicht nachvollziehbar. Da Microsoft Event-Listener für PowerPoint, wie sie im Rahmen der hier vorliegenden Arbeit verwendet werden, erst mit PowerPoint 2000 eingeführt hat [21], hätte sich die Nutzung von PowerPoint zur Aufnahme allerdings verhältnismäßig schwierig gestaltet. Auch wäre die Möglichkeit des Einsatzes von AOF unter Linux oder Unix in diesem Falle nicht gegeben gewesen. Entwicklungsgeschichtlich ist an dieser Stelle auch anzumerken, dass AOF zunächst nicht für den Betrieb unter Windows implementiert worden ist [Bacher & Ottmann 1995] und dass die der Entwicklung zugrunde liegende Philosophie sich zunächst auf die Verwendung von handschriftlichen Annotationen beschränkte [Bacher & Ottmann 1995]. Hinzu kommt, dass PowerPoint als didaktisches Werkzeug erst später ins Blickfeld der AOF-Entwickler geraten sein dürfte. Animationen, mit denen ein didaktisch sinnvoller Einsatz erst möglich wird (vgl. 2.2.2), gab es in PowerPoint beispielsweise erst ab der Version 95 [Endicott 2000].
Weitere Informationen zu AOF sowie die Möglichkeit zum Download der Software sind unter [2] und [3] zu finden. Beide Seiten scheinen nur sehr unregelmäßig gepflegt zu werden, geben aber einen guten Überblick.

## 5.3 TeleTeachingTool

Das an der Universität Trier[54] entwickelte TeleTeachingTool (TTT) zählt mit einer ersten Release-Version im Oktober 2001 zu den vergleichsweise jungen Systemen zur Vorlesungsaufzeichnung. Als Folge dieser relativ späten Entwicklung konnten bei der Konzeption des Systems einige Schwachstellen, die sich erst durch Einsatzerfahrungen mit anderen Systemen als solche herausgestellt haben, von vorneherein vermieden werden [Ziewer & Seidl 2002]. So sind sowohl die Aufnahme- als auch die Viewerkomponente des TeleTeachingTools komplett plattformunabhängig in Java implementiert. Durch die Verwendung eines VNC-Rekorders ist die Aufzeichnung beliebiger Bildschirminhalte möglich. Ein weiterer Grund für die Verwendung von VNC ist dessen verlustfreie Kompression, die Schrift und Grafiken in der Originalqualität erhält und somit deutlich zur Lesbarkeit der verwendeten Materialien beiträgt [Ziewer & Seidl 2002]. Darüber hinaus ermöglicht VNC die gleichzeitige Übertragung und Aufzeichnung eines Vortrags mit dem TeleTeachingTool.
Das System bietet die Möglichkeit graphischer Annotationen und verfügt über Volltextsuche sowie einfache folien- und zeitbasierte Navigationsfunktionen [Ziewer & Seidl 2004]. Besonders erwähnenswert ist dabei die automatische Erstellung der Indizes für Volltextsuche und Folienwechsel. Da VNC nur Bilddaten übermittelt, werden diese strukturbezogenen Daten durch ein heuristisch arbeitendes Auswertungsverfahren erschlossen [Ziewer 2004].[55] In diesem bildbasierten Verfahren werden Folienwechsel anhand der Flächengröße von Framebuffer-Änderungen detektiert.

[54] Die Entwicklung des Systems wurde zunächst an der Universität Trier begonnen, wird jedoch mittlerweile an der TU München weitergeführt [Ziewer & Seidl 2004].
[55] [Mühlhäuser & Trompler 2002] begegnen diesem Problem in einem windowsbasierten Ansatz durch die Ankopplung der in der Vorlesung verwendeten Programme an Listener über die COM-Schnittstelle der Anwendung.

Das TeleTeachingTool ist Open Source Software und kann von [25] herunter geladen werden. Ein Streaming von aufgezeichneten Dokumenten ist mit dem TeleTeachingTool nicht möglich. Die Plattformunabhängigkeit und das von der während der Vorlesung verwendeten Software unabhängige Aufnahmeverfahren stellen jedoch einen guten Grund zur ausführlicheren Betrachtung des Systems an dieser Stelle dar.

### 5.3.1 Produktionsprozess

Zur Aufzeichnung eines Vortrags mit dem TeleTeachingTool werden zwei Rechner benötigt. Zunächst muss auf dem eigentlichen Präsentationsrechner ein VNC-Server gestartet werden. Anschließend wird auf dem Aufzeichnungsrechner das TeleTeachingTool gestartet und über VNC mit dem Präsentationsrechner verbunden.[56] Abb. 5-5 zeigt die Oberfläche des TeleTeachingTools nach einem solchen Verbindungsaufbau. Das kleinere Fenster in der rechten Bildhälfte dient zur Steuerung der Aufnahme. Durch Klicken des Balkens am unteren Rand des Fensters kann die Aufnahme gestoppt und wieder gestartet werden. Dabei wird ein zusammenhängender Film erstellt, in der Aufzeichnung sind derartige Unterbrechungen lediglich durch Sprünge in Video- und Tonspur zu bemerken.
Das größere Fenster dient zur Darstellung der Bildschirmansicht des Präsentationsrechners. Dank der verwendeten VNC-Technologie können Mauszeigerbewegungen, Mausklicks und Tastendrücke direkt als Eingabesignale an den Präsentationsrechner gesendet werden. Durch Anklicken der entsprechenden Symbole am oberen Rand des größeren Fensters kann jedoch auch eine „virtuelle Glasplatte“ aktiviert werden [Ziewer & Seidl 2004]. Diese ermöglicht es, graphische Annotationen über den übertragenen Bildschirminhalt zu legen. Die Annotationen werden zusammen mit der Bildschirmübertragung aufgezeichnet und sind in das anschließend gespeicherte Dokument integriert.

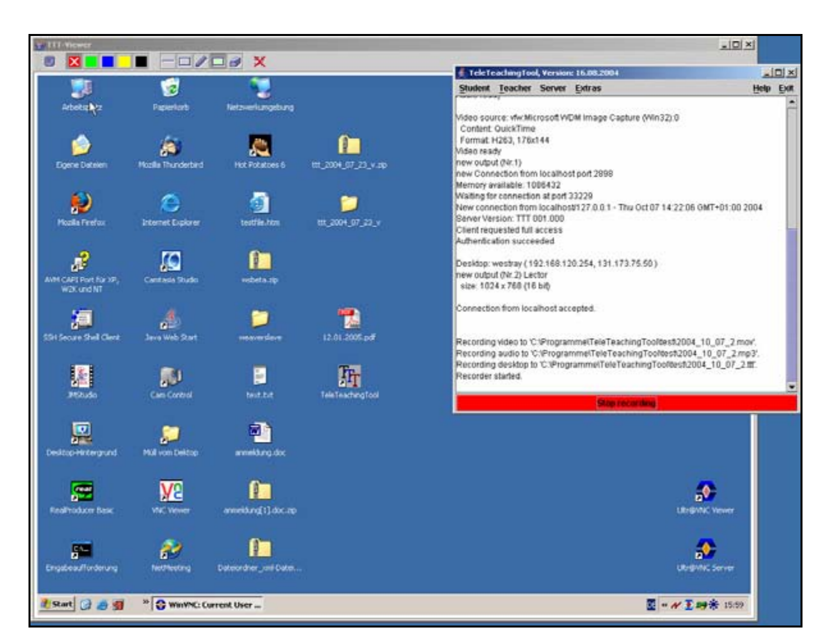



**Abb. 5-5: Ansicht des TeleTeachingTools bei der Aufzeichnung der vom Präsentationsrechner übertragenen Bildschirminhalte**

Als Nachteil erweist sich bei diesem Ansatz vor allem die durch die Übertragung bedingte Verzögerung von Mauszeigerbewegungen. Auf dem Präsentationsrechner bewegt sich die Maus dadurch sehr ruckartig und ist teilweise schlecht zu führen. Während der Verwendung der Annotationsfunktion tritt dieses Problem nicht auf, da Annotationen direkt auf dem Aufzeichnungsrechner dargestellt werden. Alternativ kann auch die Maus am Präsentations-

[56] Eine genaue Bedienungsanleitung zum Betrieb des TTT ist auf [25] zu finden.

rechner genutzt werden, dort entfällt jedoch die Möglichkeit der Nutzung der Annotationsfunktionen.
Die Notwendigkeit der Installation von VNC auf dem Präsentationsrechner und die Verteilung auf zwei Rechner macht das System nicht gerade plug and play, bei anfänglicher Betreuung der Lehrenden durch technisches Personal, sollte dies jedoch kein größeres Problem darstellen. Kamera und Mikrophon müssen an den Aufzeichnungsrechner angeschlossen werden. Bei Verbleib des Aufzeichnungsrechners im Hörsaal beschränken sich Auf- und Abbauaufwand daher auf die Verbindung des Präsentationsrechners mit dem Aufnahmerechner und evtl. den Anschluss des Präsentationsrechners an eine Stromquelle. Die Tatsache, dass das TTT auf VNC basiert, erlaubt darüber hinaus die Verwendung eines oder mehrerer Rechner, die sich nicht physikalisch im Hörsaal befinden und erhöht so die Flexibilität bei der Ausgestaltung des Vortrags.
Mit dem TTT Editor verfügt das System über die Möglichkeit, Aufzeichnungen nachzubearbeiten und beliebige Passagen herauszuschneiden [Ziewer 2007].

### 5.3.2 Studenteninterface

Für das TTT wurden drei verschiedene Studenteninterfaces entwickelt. Die ursprüngliche Viewer-Komponente des TTT öffnet zunächst eine Reihe von Fenstern, die zur Darstellung der Aufzeichnung und zur Navigation dienen (vgl. Abb. 5-6). Die Fenster können beliebig auf dem Bildschirm angeordnet werden, haben jedoch mit Ausnahme des großen Folienfensters eine fixe Größe. Die Größe des Folienfensters kann zwar verändert werden, der dargestellte Inhalt skaliert jedoch nicht mit. Größenveränderungen resultieren daher entweder in einem grauen Rand oder in der Einblendung von Scrollbars neben und unter der Foliendarstellung. Während die fixe Größe der in der Hauptansicht gezeigten Folie an dieser Stelle nur eine unerwartete Reaktion des Programms darstellt, erweist sie sich hinsichtlich der Foliennavigationsleiste (in Abb. 5-6 am rechten Bildrand zu sehen) als problematisch. Selbst auf einem 17“ TFT sind auf der Navigationsansicht der Folien nur sehr große Schriften zu erkennen, so dass die Navigationsleiste nur wenig hilfreich ist.[57] Durch Herunterregeln der Bildschirmauflösung des Abspielrechners auf die minimale Auflösung von 800x600 lässt sich die Navigationsansicht ein wenig vergrößern, wirklich praktikabel ist ein solches Vorgehen jedoch nicht. In einem neueren in [Ziewer 2007] vorgestellten Interface wird ein Framelayout verwendet, das diese Probleme beseitigt.
Die Verwendung der Volltextsuche führt zu anfangs unerwarteten Ergebnissen. Der Suchmechanismus markiert nicht etwa die Folie, auf der das Suchergebnis enthalten ist in der Navigationsansicht. Stattdessen springt die Aufzeichnung sofort zum Anfangspunkt der ersten Folie, die den gesuchten Text enthält. Wiederholtes Klicken des Suchknopfes führt zum jeweils nächsten Suchergebnis. Ein Zurückspringen zu dem Punkt der Aufzeichnung, der vor Betätigen der Suchfunktion abgespielt wurde, ist nicht möglich. Auch die Anzeige der Suchergebnisse ist auf den ersten Blick etwas ungewohnt. Bei Eingaben einzelner Buchstaben in das Suchfeld wechselt die Navigationsansicht in ein zweites Tab, das alle Folien mit passenden Suchergebnissen enthält. Bei weiteren Tastendrücken aktualisiert sich das Tab automatisch. Der Vorteil dieser Darstellungsweise ist die Fokussierung der Aufmerksamkeit auf die Suchergebnisse, der Nachteil ist der Verlust von Kontextinformationen, die bei der Suche wichtige Anhaltspunkte zur Einordnung der Folien liefern können.

[57] Erschwerend hinzu kommt die Tatsache, dass die Verwendung des TTT zur Nutzung kleinerer Schriftarten zu verführen scheint. Herkömmliche Programme zur Erstellung von Präsentationsfolien wie beispielsweise PowerPoint bieten nur eine verkleinerte Editionsansicht, in der Benutzer automatisch größere Schriftarten wählen. Die Flexibilität des TTT hinsichtlich der verwendeten Eingabeformate erweist sich daher an dieser Stelle als Schwachpunkt.

Hinzu kam in älteren Versionen des Interface die Tatsache, dass das Suchergebnis auf der erscheinenden Folie nicht markiert wird. Das Erkennen des Suchergebnisses als solches bedarf daher einigen manuellen Suchens auf der entsprechenden Folie. Gerade wenn das gesuchte Wort auf mehreren Folien innerhalb der Präsentation vorkommt, ist diese Arbeitsweise vergleichsweise mühsam. In einer neuen Version des Interfaces wird daher die Region um den erkannten Text markiert [Ziewer 2007].

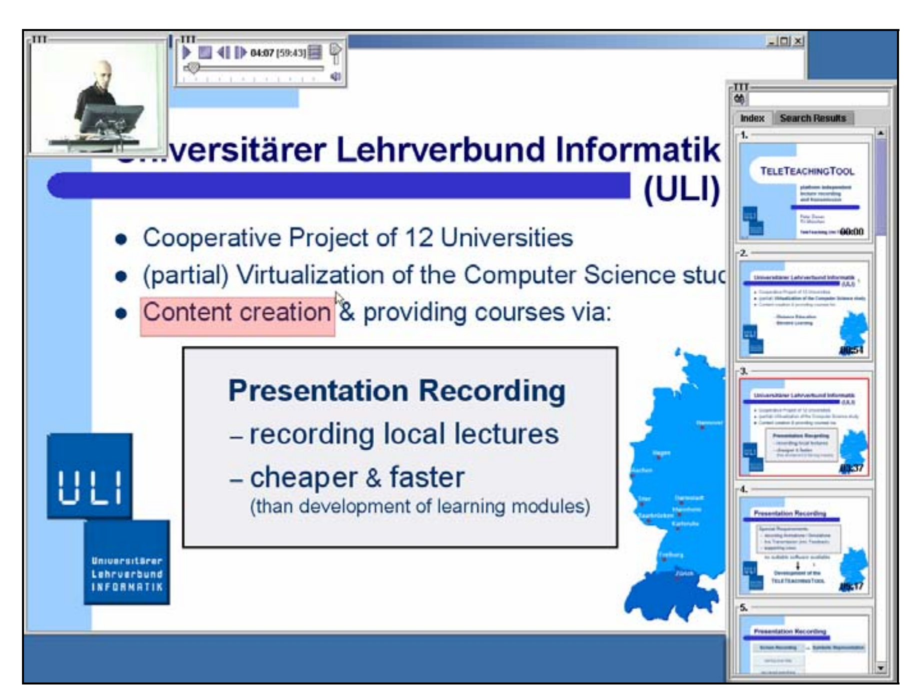



**Abb. 5-6: Älteres Studenteninterface des TeleTeachingTools**[58]

Neben der in den Player integrierten Volltextsuche existiert eine Suchmaschine für archivierte TTT-Aufzeichnungen. Die Suchmaschine bietet eine Vorschau der Folien, die den gesuchten Text enthalten und erlauben so eine Vorauswahl des gefundenen Materials. Angesichts der Tatsache, dass das TeleTeachingTool kein Streaming erlaubt und die Aufzeichnung einer Vorlesungssitzung mit Videoaufnahmen durchschnittlich größer als 100 MB ist, zeigt sich diese Vorschau als äußerst sinnvoll.

| | | | | |
|---|---|---|---|---|
| **Hypermediale Navigation** | **Allgemein** | **Backtracking** | - | |
| | | **Bookmarks** | - | |
| | | **Links** | - | |
| | | **Footprints** | - | |
| | | **Volltextsuche** | ✓ | |
| | **Strukturelle Eigenschaften** | **Inhaltsverzeichnis** | ✓ | |
| | | **„prev“-„next“-Navigation** | ✓ | **Nur in der neuen Version** |
| | | **Folienbasierte Navigation** | ✓ | |
| **Zeitbasierte Navigation** | | | ✓ | **Nur in der neuen Version** |
| **Besonderheiten** | | | **Visible scrolling** | |

**Tabelle 5-3: Navigationseigenschaften des TeleTeachingTools**

[58] Der Screenshot wurde anhand einer Aufzeichnung aus [25] erstellt. Bei der Markierung auf der dargestellten Folie handelt es sich um eine mit dem Werkzeug erstellte Annotation.

Die zeitbasierte Navigation anhand eines Sliders auf einem der kleineren Fenster lässt im Vergleich mit AOF (vgl. 5.2) oder Lecturnity [Lauer Ottmann 2002] die symbiotische Beziehung zwischen Annotationen und *random visible scrolling* deutlich werden. Annotationen lassen sich nicht über das zeitbasierte Navigationsinterface ansteuern und sind damit in diesem Zusammenhang nicht als Navigationshilfe einsetzbar. [Ziewer 2004] erwähnt allerdings die Möglichkeit eines speziellen Annotations-Events, das zum Setzen von zusätzlichen Indices verwendet werden kann. Ein in [Ziewer 2006] vorgestelltes überarbeitetes Interface unterstützt auch *random visible scrolling*.
Eine zusammenfassende Betrachtung der Navigationseigenschaften des überarbeiteten Interfaces ist in Tabelle 5-3 gegeben. Neben den beiden hier beschriebenen Interfaces existiert noch ein auf Flash basierendes Interface, das jedoch nicht alle der aufgezählten Navigationsfunktionen zur Verfügung stellt [Ziewer 2007].

### 5.3.3 Fazit und weitere Informationen

Das TeleTeachingTool ist in erster Linie als Werkzeug zur betriebssystemunabhängigen Bildschirmaufzeichnung entwickelt worden [Ziewer & Seidl 2002] und nachträglich um Annotationsmöglichkeiten [Ziewer 2004] und eine heuristische Indizierung und damit verbundene Navigationseigenschaften [Ziewer 2004] erweitert worden. Dank des Indizierungsprozesses verfügen die Aufzeichnungen über strukturelle Eigenschaften, die in Systemen zur Bildschirmaufzeichnung normalerweise nicht gegeben sind [Lauer & Ottmann 2002]. [59]
Die in Tabelle 5-3 aufgeführten Navigationseigenschaften sind daher weitaus bemerkenswerter als beispielsweise Volltextsuche in den meisten der hier behandelten Systeme. Die Möglichkeit zum Download des TeleTeachingTools, eine Vielzahl von Aufzeichnungen und weitere Informationen finden sich unter [25].

## 5.4 ePresence

Das System ePresence wird seit dem Jahr 2000 am Knowledge Media Design Institute der Universität von Toronto entwickelt [Baecker 2003].[60] Der Fokus des Projektes lag dabei zunächst auf der Entwicklung einer Lösung für so genanntes interaktives Webcasting. Das System verfügt aber auch über eine Archiv-Funktion, die zur Aufzeichnung von Veranstaltungsmitschnitten dient [Baecker, Wolf & Rankin 2004]. Interaktives Webcasting wird im Rahmen von ePresence als bezüglich der Teilnehmerzahl skalierbare Alternative zu Videokonferenzen eingesetzt [Baecker 2003].[61] Webcasting selbst ist ein Verfahren zum Internet-Broadcast von Streaming-Audio bzw. Video [Baecker 2002b]. Die Interaktivität des

---

[59] Lecturnity verfügt in eingeschränktem Maße auch über diese Möglichkeiten. Im Gegensatz zum TTT ist Indizierung und Suche dort jedoch nur in den Teilen der Aufzeichnung möglich, die als Folien in einem proprietären Format vorhanden sind. Durch eine nahtlose Aneinanderreihung von Bildschirmaufzeichnung und Folienwiedergabe können dort Foliendaten in der Aufzeichnung als strukturgebendes Element verwendet werden.
[60] Am Indian Institute of Technology (IIT) in Madras, Indien, wurde ein gleichnamiges System zur Vorlesungsaufzeichnung entwickelt. Zwischen beiden Systemen besteht jedoch keine entwicklungsgeschichtliche Beziehung. Das am IIT Madras entwickelte System verfügt nur über sehr einfache Navigationsmechanismen, beinhaltet jedoch eine Visualisierung der jeweils aktuellen Mauszeigerposition [Subrahmanyam & Raghavan 2003]. Das Studenteninterface des Systems beinhaltet ein Java-Applet und kann daher im Gegensatz zum AOF-Viewer *aofJSync* (der auch über Visualisierung der Mauszeigerposition verfügt, vgl. 5.2) online als Browseransicht zur Verfügung gestellt werden. Die Funktionalität des Studenteninterfaces ähnelt damit der des online Lecturnity-Viewers, das ePresence des IIT verfügt jedoch nicht über eine dem *random visible scrolling* verwandte Technologie.
[61] [Knaden & Zettel 2002] stellen einen verwandten didaktischen Ansatz zur Nutzung von Videokonferenzen in größeren Veranstaltungen vor.

Mediums wird bei ePresence durch die Integration eines Chats, stiller Kommunikationsmöglichkeiten zwischen einzelnen Teilnehmern, einer Fragefunktionalität und einer Moderatorenschnittstelle erreicht [Baecker 2002b]. [Baecker, Moore & Zijdemans 2003] betrachten mögliche Szenarien für den Einsatz von interaktivem Webcasting aus didaktischer Sicht.
Die Archiv-Funktion von ePresence bietet eine Reihe interessanter Navigationshilfen, wie beispielsweise Volltextsuche über alle Folieninhalte, ein Inhaltsverzeichnis und eine Zeitlinie mit Markierungspunkten für jede im Vortrag verwendete Folie [Rankin, Baecker & Wolf 2004]. Trotz seiner Doppelrolle als Webcasting- und Archivierungssystem kann und sollte ePresence daher als vollwertiges Vorlesungsaufzeichnungssystem betrachtet werden. Die mit dem System erzeugten Videomitschnitte können in den Formaten Real Media, Windows Media und MPEG4 codiert werden [Baecker, Wolf & Rankin 2004]. Die Viewer-Komponente des Systems funktioniert in allen gängigen Browsern und Betriebssystemen [Baecker, Wolf & Rankin 2004]. Eine Besonderheit des Systems ist die Möglichkeit, das Aussehen der Viewerkomponente durch selbst definierte Masken, so genannte „skins“ zu verändern [Baecker, Wolf & Rankin 2004].
Zum Jahreswechsel 2004/2005 wurde ePresence Open Source Software [Baecker, Wolf & Rankin 2004]. Bereits zuvor wurde das System auch in einigen Projekten an anderen Einrichtungen als dem Knowledge Media Institute der Universität von Toronto eingesetzt. Ein Überblick über diese Projekte ist in [Rankin, Baecker & Wolf 2004] zu finden.

### 5.4.1 Produktionsprozess

Zur Vorbereitung einer Aufzeichnung können PowerPointfolien in einem Vorbereitungsschritt konvertiert werden. Als Alternative dazu besteht die Möglichkeit, die analogen Daten, die während des Vortrags vom Präsentationsrechner zum Projektor geschickt werden, in digitale Momentaufnahmen umzuwandeln [Baecker 2003]. Darüber hinaus existiert eine Zusatzsoftware, das *Live Demo Module*, das über die *remote Desktop*-Funktionalität von Windows erlaubt, beliebige Bildschirminhalte in Form von Bildschirmaufzeichnungen mitzuschneiden [Baecker 2002a; Baecker 2003].
Da die Aufnahme der Synchronisationsdaten nicht auf dem Präsentationsrechner, sondern durch eine Reihe von Web-Services wahlweise auf einem Windows- oder Linux-Server erfolgt [Baecker, Wolf & Rankin 2004], kann der Vortragsrechner unter beliebigen Betriebssystemen gestartet werden. In dem Fall, das nicht ePresence zur Präsentation der Folien verwendet wird, müssen Folienwechsel manuell markiert werden [Baecker 2003]. Die Speicherung der Synchronisationsdaten erfolgt in einem XML-Format [Rankin, Baecker & Wolf 2004]. Mit dem *ePresence Producer* können die Daten nachbearbeitet werden, um eventuelle Synchronisationsfehler auszubessern [Rankin, Baecker & Wolf 2004]. Angesichts der Tatsache, dass Folien und Videostream während eines Live-Webcasts mit einem Zeitversatz von 5-15 Sekunden übertragen werden [Baecker 2003], kann diese Funktion durchaus hilfreich sein. Nach Beendigung der Aufnahme kann die jeweilige Aufzeichnung mit dem *ePresence Producer* direkt ins Web gestellt oder auf CD exportiert werden [Rankin, Baecker & Wolf 2004]. Beim Web-Export wird der Vortrag automatisch auf der Webseite des ePresences-Projektes eingefügt [Baecker, Wolf & Rankin 2004]. Zur inhaltlichen Nachbearbeitung erlaubt es der *ePresence Producer*, manuell Suchbegriffe zu einzelnen Kapiteln oder Folien hinzuzufügen. Bei Verwendung von PowerPoint wird der Folientext automatisch in die Suchwortbasis übernommen [Baecker, Wolf & Rankin 2004]. Die Einbindung von Spracherkennungssoftware ist von den Entwicklern angedacht [Rankin, Baecker & Wolf 2004].
Durch den Einsatz mehrerer Rechner zur Aufnahme des Videomitschnittes ist in ePresence eine parallele Codierung in unterschiedliche Videoformate möglich [Rankin, Baecker &

Wolf 2004]. Mittels der Verwendung von selbst- oder vordefinierten *templates* oder *skins* können darüber hinaus auch individuell gestaltete Benutzermasken zur Wiedergabe von Aufzeichnungen erzeugt werden. In einem durch das Aufzeichnungsformat begrenzten Rahmen kann die Wiedergabemaske so an technische und kursspezifische Anforderungen angepasst werden. Die Umwandlung der aufgezeichneten Rohdaten in die gewünschte Benutzermaske geschieht dabei auf der Basis von XSLT [Baecker 2003]. Zur Erstellung eines eigenen *templates* werden jedoch auch Kenntnisse in der Programmierung von HTML und verschiedenen Scriptsprachen vorausgesetzt [Baecker, Wolf & Rankin 2004]. Für technisch unbedarfte Lehrende dürfte die Gestaltung eigener *templates* daher in den meisten Fällen ausgeschlossen sein.

### 5.4.2 Studenteninterface

Die Tatsache, dass ePresence die Möglichkeit zur individuellen Gestaltung der Benutzermaske des Studierendeninterfaces durch die oben genannten *templates* bietet, mag die Diskussion des Interfaces an dieser Stelle auf den ersten Blick in Frage stellen. Der Fokus der hier durchgeführten Betrachtung liegt jedoch nicht auf dem Design der Interfaces, sondern auf deren Navigationsfunktionalität. Da die Erstellung eines *templates* erheblicher Sachkenntnis bedarf, ist in der Regel ohnehin von der Verwendung des Standard-*templates* auszugehen. Ein Blick in die Archive des ePresence-Systems [32] bestätigt diese Vermutung. Die Verwendung unterschiedlicher *templates* scheint dort lediglich zur Anpassung an die Vortragsform wie beispielsweise einen Vortrag ohne Folien zu dienen (siehe Abb. 5-7). Ein weiterer Einsatzpunkt der Technologie ist die Einbindung von Präsentationen in ein Corporate Design Muster. Da in keinem dieser Fälle die Navigationsmerkmale des Studierendeninterfaces verändert werden, kann die folgende Betrachtung also als exemplarisch für das System angesehen werden.
Ein weiteres Argument zur Analyse der Standard-Benutzermaske ist die Tatsache, dass ePresence eine Reihe von Besonderheiten aufweist, die so nicht in anderen Systemen zu finden sind und damit einen weiteren Stein im Mosaik teilweise hypermedial navigierbarer Vorlesungsaufzeichnungsinterfaces darstellt.

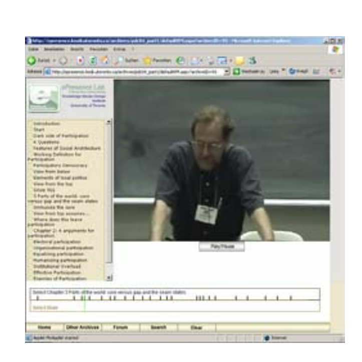







**Abb. 5-7: Verschiedene Ausprägungen der ePresence Benutzeroberfläche (von links nach rechts: Vortrag ohne Folien, Gestaltung der Oberfläche um 2002, Gestaltung der Oberfläche um 2004)**

Das standardmäßig von ePresence erzeugte Studierendeninterface besteht aus mehreren HTML-Frames und einem Java-Applet [Baecker 2003]. Bei genauerer Betrachtung des Quellcodes erweist sich die *timeline*, eine interaktive Zeitlinie (Abb. 5-8) als dieses Java-Applet. Die *timeline* ist in zwei Bereiche eingeteilt, die durch eine horizontale Trennlinie voneinander abgegrenzt werden. Oberhalb der Trennlinie sind jeweils die Markierungspunkte für den Anfang eines Kapitels zu finden, unterhalb der Linie werden die Zeitpunkte

der einzelnen Folienwechsel angezeigt (vgl. 5.4.1). Beim Überfahren dieser Markierungen mit dem Mauszeiger wird je nach Auswahl direkt neben dem Text *Select Chapter:* bzw. *Select Slide:* am linken Rand der Zeitlinie der Titel der ausgewählten Folie bzw. des ausgewählten Kapitels angezeigt. Durch Anklicken einer beliebigen Stelle innerhalb der *timeline* wird die Aufzeichnung an der gewählten Position beginnend abgespielt. Die aktuelle Abspielposition der Aufzeichnung ist auf der *timeline* durch eine durchgehende vertikale grüne Linie gekennzeichnet.

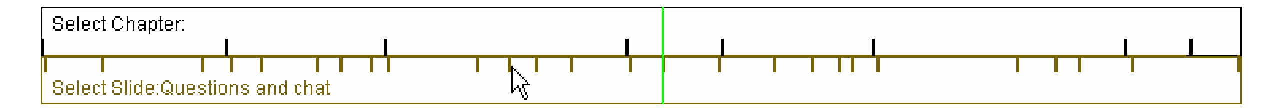



**Abb. 5-8: Die interaktive *timeline* des ePresence Systems**

Durch die Anwahl des Suchknopfes am unteren Rand der Benutzermaske springt das Applet in einen Suchmodus (siehe Abb. 5-9). In der Suchmaske befindet sich der Suchknopf dann direkt neben dem Eingabefeld. Nach Abschicken der Suchanfrage springt das System zurück in den *timeline*-Modus und zeigt dort alle Vorkommnisse des Suchbegriffes durch jeweils eine vertikale blaue Linie an den entsprechenden Positionen an. Dabei werden nicht nur die Anfangspunkte der Kapitel und Folien angezeigt, für die der gesuchte Begriff in der Suchwortbasis gefunden wurde. Für den Fall, dass bei der manuellen Eingabe des Suchbegriffes im *ePresence Producer* ein entsprechender Eintrag vorgenommen wurde, kann auch ein beliebiger Zeitpunkt markiert werden [Baecker 2003].

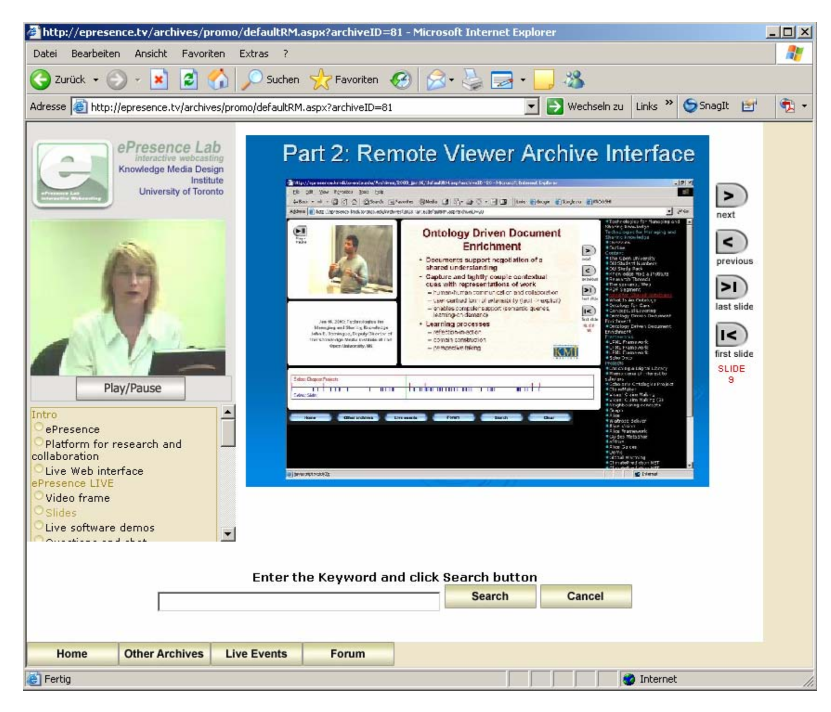



**Abb. 5-9: ePresence im Suchmodus**

Der Wechsel zwischen Such- und Anzeigemodus zeigt sich bei der Stichwortsuche teilweise als recht umständlich. Dieses Vorgehen ist besonders dann hinderlich, wenn nicht auf Anhieb der richtige Suchbegriff gewählt wird und so immer wieder die Suchmaske aufgerufen werden muss. Zusätzlich zur Suchfunktionalität innerhalb einer Aufzeichnung existiert auch eine archivweite Suche, die eine Linkliste aller Aufzeichnungen, in denen der entsprechende Suchbegriff vorkommt, als Ergebnis liefert. Beim Anklicken eines solchen Links wird die gewählte Aufzeichnung geöffnet. Der Suchbegriff wird mit der URL übergeben, so dass die entsprechenden Stellen in der *timeline* bereits markiert sind.

Durch Anklicken der Foliennavigationselemente (in Abb. 5-9 am rechten Bildrand zu sehen) kann asynchron zum Vortrag im Foliensatz der Aufzeichnung navigiert werden. Diese Funktionalität ist vor allem für Live-Übertragungen als Äquivalent zum Blättern in einem Skript konzipiert worden [Baecker 2003]. Beim Übergang des Films in einen neuen Folienabschnitt wird wieder die jeweils aktuelle Folie eingeblendet.
Implementationsbedingt wirkt sich Klicken des Back-Knopfes des verwendeten Browsers auf die angezeigte Folie aus. Mit Hilfe des browser-eigenen Backtrackings kann daher durch die Reihenfolge der dargestellten Folien navigiert werden. Die Abspielposition des Films wird dadurch jedoch nicht beeinflusst.
Weitere Navigationshilfen sind nicht in das Interface integriert. Tabelle 5-4 gibt einen abschließenden Überblick aller im System enthaltenen Navigationseigenschaften.
Negativ auffallend sind die schlechte Bildqualität der Folien in vielen der im Archiv enthaltenen Aufzeichnungen, die aus dem Export der Folien aus PowerPoint resultiert und die fehlende Skalierbarkeit des Interfaces. Besonders störend ist letzteres bei der *timeline*, die beim Öffnen des Interfaces in einem kleineren Browserfenster unbemerkt über den rechten Fensterrand hinaus verläuft. Als Konsequenz werden beispielsweise auch Suchergebnisse im hinteren Teil der Aufzeichnung nicht im sichtbaren Bereich angezeigt.

| | | | | |
|---|---|---|---|---|
| **Hypermediale Navigation** | **Allgemein** | **Backtracking** | - | |
| | | **Bookmarks** | - | |
| | | **Links** | - | **Die Möglichkeit Suchbegriffe in der URL zu übergeben kann jedoch bei entsprechender Manipulation zweckentfremdet werden.** |
| | | **Footprints** | - | |
| | | **Volltextsuche** | ✓ | **Es existiert zusätzlich eine externe Suchmaschine** |
| | **Strukturelle Eigenschaften** | **Inhaltsverzeichnis** | ✓ | |
| | | **„prev“-„next“-Navigation** | - | **Die entsprechenden Navigationselemente dienen der asynchronen Navigation in den Folien** |
| | | **Folienbasierte Navigation** | ✓ | **über die interaktive *timeline*** |
| **zeitbasierte Navigation** | | | ✓ | |
| **Besonderheiten** | | | **interaktive *timeline*** | |

**Tabelle 5-4: Navigationseigenschaften des ePresence Benutzerinterfaces**

### 5.4.3 Fazit und weitere Informationen

Das System ePresence ist in erster Linie als Werkzeug zum interaktiven Webcasting entwickelt worden. Dass Aufzeichnungen dabei als Nebenprodukt einer Übertragung anfallen, wirkt sich jedoch kaum negativ auf die Qualität der Aufzeichnungen aus. Einzig das Fehlen einer Animationsbasierten Foliensynchronisation ist an dieser Stelle zu bemerken. Bei einfachen Animationen scheint die von den Entwicklern des Systems favorisierte Lösung die manuelle Zerlegung der Animationsschritte auf mehrere Folien zu sein.
Bis auf die Implementation der Suchfunktion sind die Navigationselemente des Systems recht gut durchdacht. Besonders die interaktive *timeline* und die Tatsache, dass dort in einem Navigationselement Strukturinformationen aus den beiden Ebenen Kapitel und Folie zusammengefasst werden, ist hier ein Pluspunkt. Auch die Verbindung von Strukturinformationen und dem zeitlichen Ablauf der Vorlesung ist mit der *timeline* gelungen.

Der aktuelle Entwicklungsstand des Systems wir von den Entwicklern zeitnah in einem Blog dokumentiert [5].
Die Webseite des Projektes ist [31]. Dort sind neben generellen Informationen auch alle archivierten Aufzeichnungen zu finden.

## 5.5 eClass/Classroom 2000

Das Projekt Classroom 2000 wurde im Juli 1995 am Georgia Institute of Technology in Atlanta, Georgia (USA) begonnen [Abowd 1999]. Es gehört damit zusammen mit AOF zu den frühesten Forschungsprojekten im Bereich Vorlesungsaufzeichnungssysteme [Belenkaia, Lienhard, Mohamed & Ottmann 2004]. Zielsetzung war die Entwicklung eines Systems zur Aufzeichnung von Vorträgen und Meetings und zur Anreicherung der Aufzeichnung mit verschiedenen Medien [Abowd 1999]. Am 1. 1. 2000 wurde das Projekt in eClass umbenannt [Graca Pimentel, Ishiguro, Abowd, Kerimbaev & Guzdial 2001]. 2001 wurde das Projekt eingestellt [Friedland 2006]. In [Brotherton & Abowd 2004] wird im Rückblick eine Analyse der mit dem System gesammelten Erfahrungen gegeben.
Im Rahmen von eClass ist eine Reihe von wegweisenden Entwicklungen entstanden. So ist beispielsweise das in 4.2.1.3 beschriebene linking by interacting Paradigma als Bestandteil des Classroom 2000 entwickelt worden. Bereits in den späten 90er Jahren wurde mit dem System ein Wiki-ähnlicher Ansatz realisiert, in dem Studierende und Lehrende mit dem System erstellte Aufzeichnungen nachträglich kommentiert werden können [Abowd, Graca Pimentel, Ishiguro, Kerimbaev & Guzdial 1999]. Auch technisch spielte das System seinerzeit eine Vorreiterrolle. Obwohl die im Verlauf des Projektes erstellten Aufzeichnungen heute nicht mehr auf der Projekthomepage zugänglich sind, wird daher im Folgenden ein kurzer Überblick über das System gegeben.

### 5.5.1 Produktionsprozess

Zur Aufzeichnung der Lehrmedien wird in eClass ein elektronisches Whiteboard verwendet [Brotherton 2001]. Auf dem Whiteboard können Tafelanschrieb und vorbereitete Materialien angezeigt und aufgezeichnet werden. Zur Aufzeichnung dient die in Java implementierte Softwarekomponente ZenPad, die den Tafelanschrieb speichert und rudimentäre Präsentationsoptionen für in das System importierte Bilder bietet. Zusätzlich kann ein Webbrowser auf dem Whiteboard angezeigt werden.

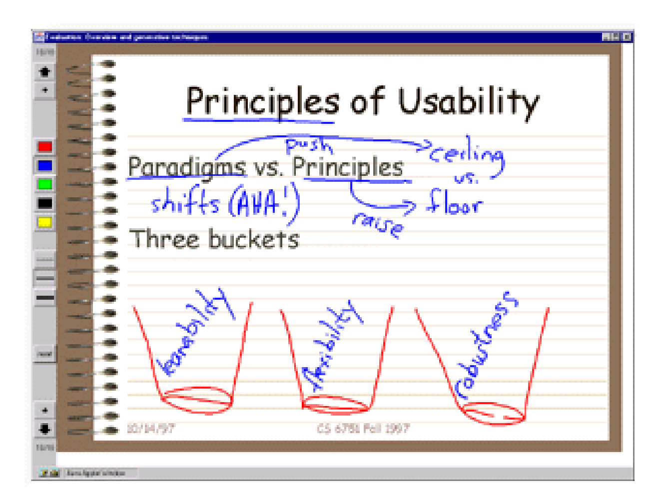



**Abb. 5-10: Ansicht des eClass Whiteboard-Interfaces**[62]

[62] Entnommen aus [30].

Zur Protokollierung der Adressen besuchter Webseiten werden alle Web-Verbindungen durch einen Proxy-Server geleitet.[63] Zum Import von PowerPoint-Präsentationen kann die Komponente TransferMation genutzt werden, die eine Präsentation in statische Bilder umwandelt. Animationen gehen dabei verloren. Audio und Video werden auf einem Rechner direkt in die Formate RealAudio und RealVideo encodiert. Zur Aufzeichnung des Videos wird dabei (vermutlich aus Performance-Gründen) eine niedrige Aufzeichnungsqualität verwendet. Die Aufzeichnung aller Streams einschließlich der im ZenPad aufgezeichneten Daten wird zentral durch einen Server (den ZenMaster) koordiniert. Im Anschluss an die Aufzeichnung werden die Streams anhand von Zeitstempeln synchronisiert. In einem Post-editing-Interface kann die Aufzeichnung darüber hinaus noch mit weiteren Informationen wie einem manuellen Transcript von handschriftlichen Anmerkungen versehen werden. Dieser Weg wurde gewählt, da sich eine automatische Handschrifterkennung im Rahmen des Projektes aufgrund der häufigen Kombination von Freihandzeichnungen und Handschrift nicht realisieren ließ [Graca Pimentel, Ishiguro, Abowd, Kerimbaev & Guzdial 2001].

### 5.5.2 Studenteninterface

Zur Wiedergabe von mit eClass aufgezeichneten Mitschnitten wurden mehrere verschiedene Interfaces implementiert, die in [Brotherton 2001] detailliert beschrieben sind. Eine wesentliche Navigationsfunktion, die in mehreren der Interfaces implementiert wurde, ist das Linking by Interacting Paradigma [Graca Pimentel, Abowd & Ishiguro 2000]. Mit dieser Technik werden handschriftliche Anmerkungen des Dozenten während des Vortrags (vgl. Abb. 5-10) in einer zusätzlichen Ebene über die Ansicht der Folie gelegt und einzeln verlinkt. Durch Anklicken einer Figur oder eines Schriftzugs kann dann direkt der Zeitpunkt in der Aufzeichnung annavigiert werden, an dem die Figur bzw. der Schriftzug erstellt wurde. Darüber hinaus verfügten die Interfaces über zeitbasierte Navigation in Form einer Zeitleiste und teilweise Folienüberblicke in Form einer Thumbnailübersicht. Im sogenannten *Toilet-Roll-Interface* wurde die Anzeigedauer von Folien in der Aufzeichnung in der Zeitleiste visualisiert (siehe Abb. 5-11). Die roten Markierungen in der grünen Leiste heben Abschnitte hervor, in denen handschriftliche Anmerkungen vorgenommen wurden.

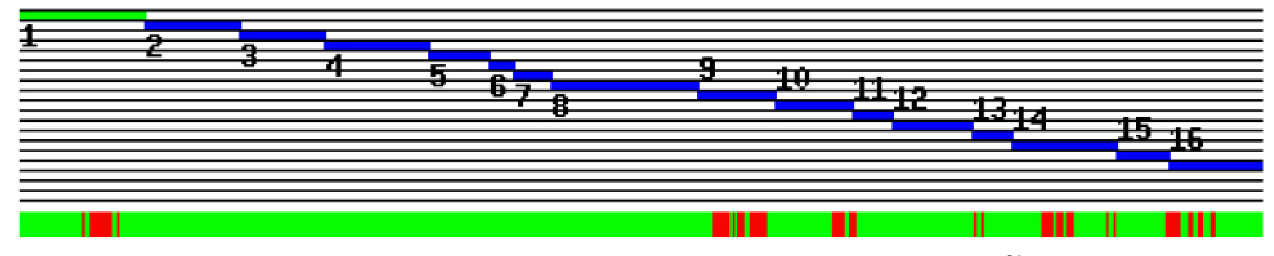



**Abb. 5-11: *Toilet-Roll-Interface* des Studierendeninterfaces von eClass**[64]

Zu einzelnen Folien konnten durch Lehrende oder Studierende kollaborativ Anmerkungen erstellt werden. Mittels einer Suchmaske konnte der Folientext, vom Dozenten in symbolischer Form ergänzte Transcripte handschriftlicher Anmerkungen (vgl. 5.5.1), die kollaborativen Anmerkungen, besuchte Webseiten, Anmerkungen des Dozenten und in einigen Fällen auch die Sprachspur der Aufzeichnung durchsucht werden.
Auch zur Suche über alle Aufzeichnungen eines Kurses existierte eine Suchmaschine. Die Vielzahl der implementierten Interfaces war vornehmlich durch Browserkompatibilitäts-

[63] Wie im Rahmen des Einsatzes anderer Systeme festgestellt, ist eine derartige Protokollierung besuchter Webadressen nur bedingt sinnvoll, da die besuchten Webseiten im Nachhinein auf den jeweiligen Servern geändert oder gelöscht werden können.
[64] Entnommen aus [Brotherton 2001].

probleme begründet. Ein Überblick zu den Navigationsfunktionen von eClass ist in Tabelle 5-5 gegeben.

| | | | | |
|---|---|---|---|---|
| **Hypermediale Navigation** | **Allgemein** | **Backtracking** | - | |
| | | **Bookmarks** | - | |
| | | **Links** | ✓ | **Über die kollaborative Anmerkungsfunktion (CoWeb)** |
| | | **Footprints** | - | |
| | | **Volltextsuche** | ✓ | **Es existiert zusätzlich eine externe Suchmaschine** |
| | **Strukturelle Eigenschaften** | **Inhaltsverzeichnis** | ✓ | |
| | | **„prev"-„next"-Navigation** | - | |
| | | **Folienbasierte Navigation** | ✓ | |
| **Zeitbasierte Navigation** | | | ✓ | |
| **Besonderheiten** | | | **Linking by Interacting** | |

**Tabelle 5-5: Navigationseigenschaften der eClass Benutzerinterfaces**

### 5.5.3 Fazit und weitere Informationen

Das Projekt eClass/Classroom 2000 war eines der frühesten Projekte zur elektronischen Aufzeichnung von Vorlesungen. Im Rahmen des Projektes wurden viele Entwicklungen erarbeitet, die auch heute noch wegweisend für Vorlesungsaufzeichnungssysteme sind. Die Homepage des Projektes ist unter [30] zu finden und enthält neben der Beschreibung des Projektes auch ein umfangreiches Literaturverzeichnis.
Obwohl das Projekt 2001 eingestellt wurde, sind Weiterentwicklungen wie das in [Cooperstock 2003] beschriebene System auch heute noch im Einsatz.

## 5.6 eTEACH

Das System eTEACH wird an der Universität von Wisconsin in Madison, Wisconsin entwickelt und wurde dort zum ersten Mal im Jahr 2000 eingesetzt [Almeida 2003]. Das System ist kein Vorlesungsaufzeichnungssystem im herkömmlichen Sinne, sondern eher ein Autorensystem zur Erstellung multimedialer Lerninhalte [Moses, Litzkow, Foertsch & Strikwerda 2002]. In der Produktion von Lernmaterialien ist eTEACH deutlich komplizierter zu handhaben als andere Systeme zur Vorlesungsaufzeichnung. Nichtsdestotrotz ist die Studierendensicht auf das System mit den Studierendeninterfaces anderer Systeme zur Vorlesungsaufzeichnung vergleichbar.
Eine im Rahmen dieser Arbeit relevante Besonderheit des Systems ist die Implementation einer Footprint-Funktionalität. Ein weiteres Alleinstellungsmerkmal des Systems ist ein besonderer Fokus auf Barrierefreiheit. Das System unterstützt als einziges seiner Art Bildschirmleseprogramme und Untertitel [Litzkow, Anderson, Bundy & Ewers 2004].
Technisch fällt an eTEACH auf, dass die aufgezeichneten Inhalte nur im Microsoft Internet Explorer wiedergegeben werden können. Dieses Vorgehen ermöglicht die korrekte Visualisierung von PowerPoint-Animationen im aufgezeichneten Material und eine hochqualitative Darstellung der verwendeten Folien. Zusammen mit einer brauchbaren Navigationsstruktur ermöglichen diese Faktoren den Einsatz von eTEACH-Aufzeichnungen als vollwertiges Lehrmaterial. So wurde mit eTEACH bereits wiederholt ein neuartiges didaktisches Szenario erfolgreiche durchgeführt, in dem Vorlesungen komplett durch Aufzeichnungen ersetzt worden sind [Foertsch, Moses, Strikwerda & Litzkow 2002].

## 5.6.1 Produktionsprozess

Die technischen Voraussetzungen zum Einsatz des eTEACH-Autorenwerkzeugs umfassen einen Windows-Rechner mit Sound- und Grafikkarte. Benötigte Zusatzsoftware wie spezielle Video und Audio-Codecs, die von eTEACH benötigt werden [Bijnens, Bijnens & Vanbuel 2004], werden in neueren Versionen der Software mitgeliefert, so dass keine zusätzlichen Installationsschritte nötig sind.

Die Produktion einer Lerneinheit mit dem eTEACH-Autorenwerkzeug ist im Vergleich mit den anderen hier beschriebenen Systemen sehr aufwendig. Der Grund dafür liegt vor allem in der Tatsache, dass in eTEACH alle Arbeitsschritte sequenziell hintereinander ausgeführt werden müssen. So wird die Filmaufzeichnung in eTEACH nicht automatisch während der Aufnahme mit der Folienpräsentation synchronisiert, sondern erst in einem nachträglichen Arbeitsgang mit dem Folienmaterial verknüpft [Moses, Litzkow, Foertsch & Strikwerda 2002]. Der Arbeitsaufwand zur Erstellung einer fertigen Aufzeichnung wird damit mehr als verdoppelt, da zunächst das zu verwendete Video gedreht werden muss und im Anschluss an die manuell durchgeführte Synchronisation eine Überprüfung der Web-Präsentation stattfinden sollte. In Anbetracht von Erfahrungen, die in den vergangenen Jahren am Zentrum zur Unterstützung virtueller Lehre der Universität Osnabrück gemacht worden sind, beträgt der zeitliche Aufwand zur manuellen Synchronisation noch einmal mindestens die Dauer der Videoaufzeichnung. In der Regel ist damit also sogar von einer Verdreifachung des Arbeitsaufwands auszugehen.

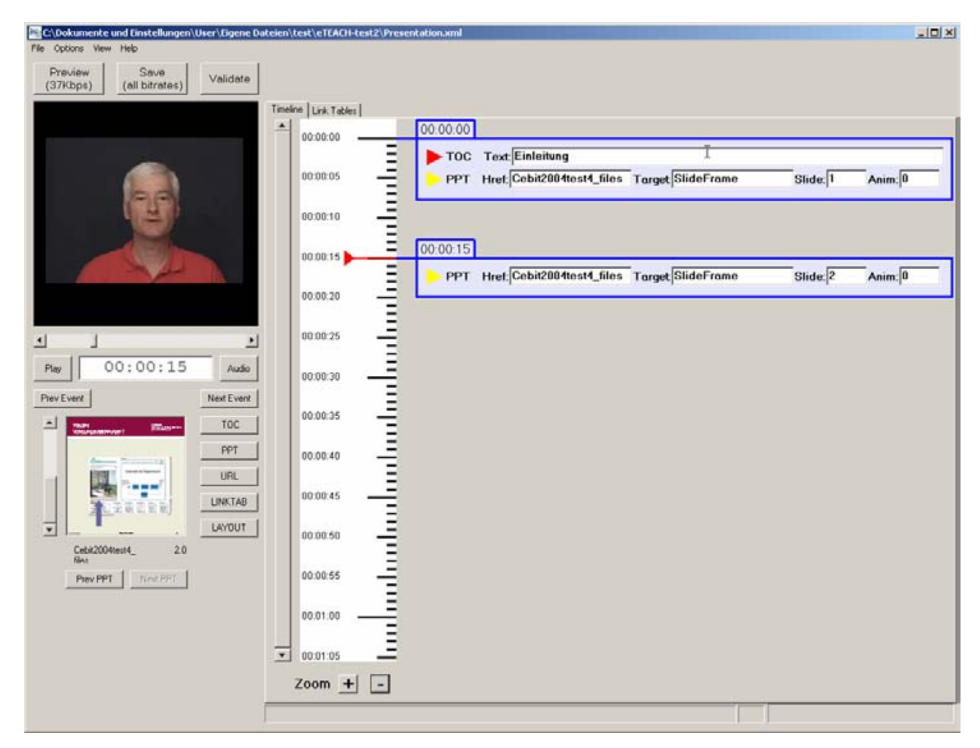



**Abb. 5-12: Das eTEACH-Autorenwerkzeug**

Abb. 5-12 zeigt das eTEACH-Autorenwerkzeug, mit dem die Synchronisation der verschieden Quellen wie Video und Folienmaterial durchgeführt werden kann. Hier können auch Einstiegspunkte für ein anklickbares Inhaltsverzeichnis und Links auf externes Material definiert werden [Moses, Litzkow, Foertsch & Strikwerda 2002]. Etwas unintuitiv erscheint die Tatsache, dass Einträge ins Inhaltsverzeichnis nicht automatisch mit dem Beginn einer neuen Folie generiert werden sondern manuell erstellt werden müssen. Da das Folienvorschaufenster in der linken unteren Ecke des Anwendungsfensters keine symbolische Repräsentation der Folientexte enthält, können Folientitel auch nicht von dort kopiert

werden. Davon abgesehen ist das Autorenwerkzeug recht einfach zu bedienen. Um die Kontrolle einer erstellten Web-Präsentation zu erleichtern, bietet das System einen Vorschaumodus.
Die Stärken und Schwächen von eTEACH zeigen sich beim Import von PowerPointpräsentation und später auch bei deren Wiedergabe. Der Import von einfach gehaltenen PowerPointdateien stellt für das System kein Problem dar. Auch komplizierte Effekte wie Animationspfade (vgl. 8.1.1.2.1.1) und Hervorheben (vgl. 8.1.1.2.1.2) werden vom System verarbeitet. Sogar Nacheffekte (vgl. 8.1.1.2.3.1) werden ohne Komplikationen importiert.
Probleme treten jedoch bei dem Versuch auf, Folien zu importieren, auf denen ein und dasselbe Objekt mehrfach animiert wird. An dieser Stelle bricht das System den Import der Folien mit einer Fehlermeldung ab. Folien, in denen etwa ein komplizierter technischer Sachverhalt durch mehrmalige Animation einer Komponente dargestellt werden soll, sind daher im System nicht verwendbar.
Zur Wiedergabe von PowerPoint-Animationen in der späteren Web-Präsentation wird offensichtlich ein an den HTML-Export von PowerPoint angelehnter Mechanismus genutzt. Die meisten Effekte werden daher genau so wiedergegeben wie sie auch in der Originalpräsentation vorkommen. Alternativ zu PowerPoint-Dateien können auch beliebige HTML-Dateien importiert werden [Moses, Litzkow, Foertsch & Strikwerda 2002], wobei die Mechanismen zur Synchronisation der verschiedenen Medien unverändert bleiben. Die Mehrzahl der Benutzer verwendet jedoch importierte PowerPointfolien [Moses, Litzkow, Foertsch & Strikwerda 2002].
Im Anschluss an die Fertigstellung einer Präsentation kann das Material den Studierenden sowohl im Internet als auch auf CD zugänglich gemacht werden [Foertsch, Moses, Strikwerda & Litzkow 2002].

### 5.6.2 Studierendeninterface

Die Web-Ausgabe des eTEACH-Autorenwerkzeugs ist für den Microsoft Internet Explorer konzipiert [Moses, Litzkow, Foertsch & Strikwerda 2002]. Bei dem Versuch, eTEACH-Präsentationen mit anderen Browsern wie Mozilla oder dem Netscape Navigator zu öffnen erscheint ein Warnfenster, das über die unzureichende Eignung des Browsers informiert.
Im Internet Explorer wird im Folienfenster des eTEACH-Studierendeninterfaces eine direkt aus PowerPoint exportierte HTML-Ansicht der verwendeten Folien gezeigt. Durch dieses Vorgehen wird sowohl eine hervorragende Bildqualität als auch eine fehlerfreie Wiedergabe der angegebenen Animationsreihenfolge ermöglicht. Besonders der letztere dieser beiden Faktoren hat beim erfolgreichen Einsatz des Systems in neuartigen didaktischen Szenarien eine entscheidende Rolle gespielt [Moses, Litzkow, Foertsch & Strikwerda 2002]. Das verwendete Video wird gestreamt. Die Wiedergabe einer Aufzeichnung ist daher ohne längere Wartezeiten oder größeren Speicherplatzbedarf auf der Festplatte möglich.
Die Navigationsmöglichkeiten des Systems sind im Vergleich mit anderen Systemen nicht besonders weit entwickelt, das System ist jedoch das einzige Vorlesungsaufzeichnungssystem, das eine Footprint-Funktionalität implementiert. Über das textuelle Inhaltsverzeichnis des Systems wird angezeigt, welcher Abschnitt der Aufzeichnung momentan abgespielt wird und welche Abschnitte der Nutzer bereits angesehen hat. Der aktive Abschnitt wird dabei fett, die bereits besuchten Inhalte durchgestrichen markiert (siehe Abb. 5-13). Zur Markierung bereits besuchter Inhalte wird automatisch jeder Abschnitt markiert, der einmal aktiviert war. Nachteil dieser Vorgehensweise ist, dass auch Abschnitte, die nur wenige Sekunden abgespielt worden sind, als bereits besucht markiert werden. Ein weiterer Nachteil ist die fehlende Persistenz der Markierungen über erneute Besuche der Seite hinweg. Durch Verlassen der Seite sind die Footprint-Informationen damit verloren.

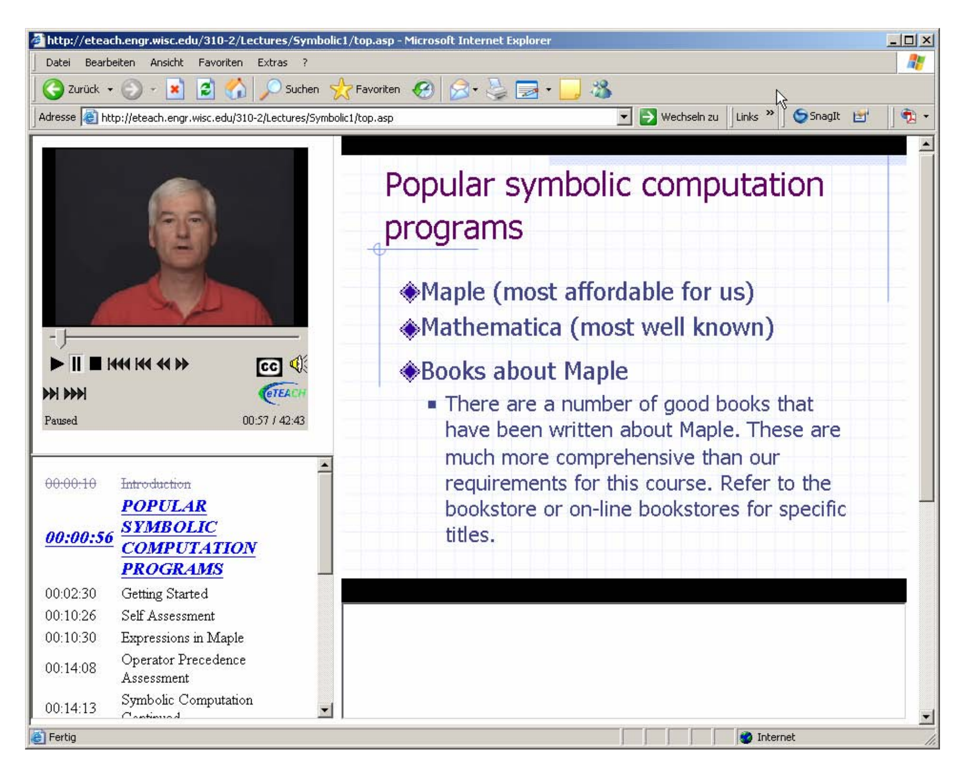



**Abb. 5-13: Das Studierendeninterface des eTEACH-Systems**

Neben Ihrer Eigenschaft als Footprints haben die Einträge im Inhaltsverzeichnis auch eine klassische Navigationsfunktion. Durch Anklicken eines Titels mit der Maus kann jeweils direkt zum Anfangspunkt des bezeichneten Abschnitts navigiert werden. Zur zeitbasierten Navigation können die Knöpfe und der Slider im Videobedienteil des Interfaces genutzt werden. Die Knöpfe erlauben das Vorwärts- und Rückwärtsspringen anhand fester Zeitintervalle (10 und 30 Sekunden) sowie ein beschleunigtes Abspielen des Films und sind im Anschluss an eine Nutzerbefragung in das Interface integriert worden [Moses, Litzkow, Foertsch & Strikwerda 2002].
Ein abschließender Überblick über alle in eTEACH implementierten Navigationselemente wird in Tabelle 5-6 gegeben.

<table>
<tr><td rowspan="8">Hypermediale Navigation</td><td rowspan="5">Allgemein</td><td>Backtracking</td><td>-</td><td></td></tr>
<tr><td>Bookmarks</td><td>-</td><td></td></tr>
<tr><td>Links</td><td>-</td><td></td></tr>
<tr><td>Footprints</td><td>✓</td><td></td></tr>
<tr><td>Volltextsuche</td><td>-</td><td></td></tr>
<tr><td rowspan="3">Strukturelle Eigenschaften</td><td>Inhaltsverzeichnis</td><td>✓</td><td></td></tr>
<tr><td>„prev“-„next“-Navigation</td><td>-</td><td>Die entsprechenden Navigationselemente dienen zur zeitbasierten Navigation</td></tr>
<tr><td>Folienbasierte Navigation</td><td>✓</td><td></td></tr>
<tr><td colspan="3">Zeitbasierte Navigation</td><td>✓</td><td></td></tr>
<tr><td colspan="3">Besonderheiten</td><td colspan="2">Navigation anhand fester Zeitintervalle, beschleunigter Vor- und Rücklauf</td></tr>
</table>

**Tabelle 5-6: Navigationseigenschaften des eTEACH Benutzerinterfaces**

### 5.6.3 Fazit und weitere Informationen

Das System eTEACH hebt sich vor allem durch seine Implementation der Footprint-Eigenschaft, eine sehr gute Folienqualität und eine korrekte Wiedergabe der Animationsreihenfolge der verwendeten Effekte positiv von der Masse ab. Negativ fallen der zeitaufwändige Produktionsprozess und Probleme mit Mehrfachanimationen auf.
Das System ist wiederholt mit Erfolg in der Lehre eingesetzt worden, was wohl nicht zuletzt auf den oben genannten positiven Eigenschaften beruht.
Weiterführende Informationen zum System, eine größere Anzahl von Aufzeichnungen und die Möglichkeit zum Download der Produktionssoftware ist unter [10] zu finden.

## 5.7 Fazit und Überblick über alternative Navigationsansätze

Neben den bereits in diesem Kapitel beschriebenen Systemen existiert noch eine Vielzahl von weiteren Systemen, die vereinzelt über hypermediale Navigationsfunktionen verfügen. Dieser Abschnitt gibt einen Überblick über die Verteilung hypermedialer Navigationsfunktionen in anderen Systemen und ordnet die in 5.1 bis 5.6 beschriebenen Systeme in diesen Überblick ein. Darüber hinaus werden kurz eine Reihe alternative Navigationsansätze vorgestellt.

### 5.7.1 Überblick und Einordnung

Wie bereits in 4.2.2.1 dargestellt, können Navigationsansätze für Vorlesungsaufzeichnungen entlang der Achsen zeit- und strukturbasierte Navigation eingeordnet werden. In Abb. 5-14 wird an diese Systematik angelehnt ein Überblick über die bereits in diesem Kapitel beschriebenen und einige weitere Systeme gegeben.

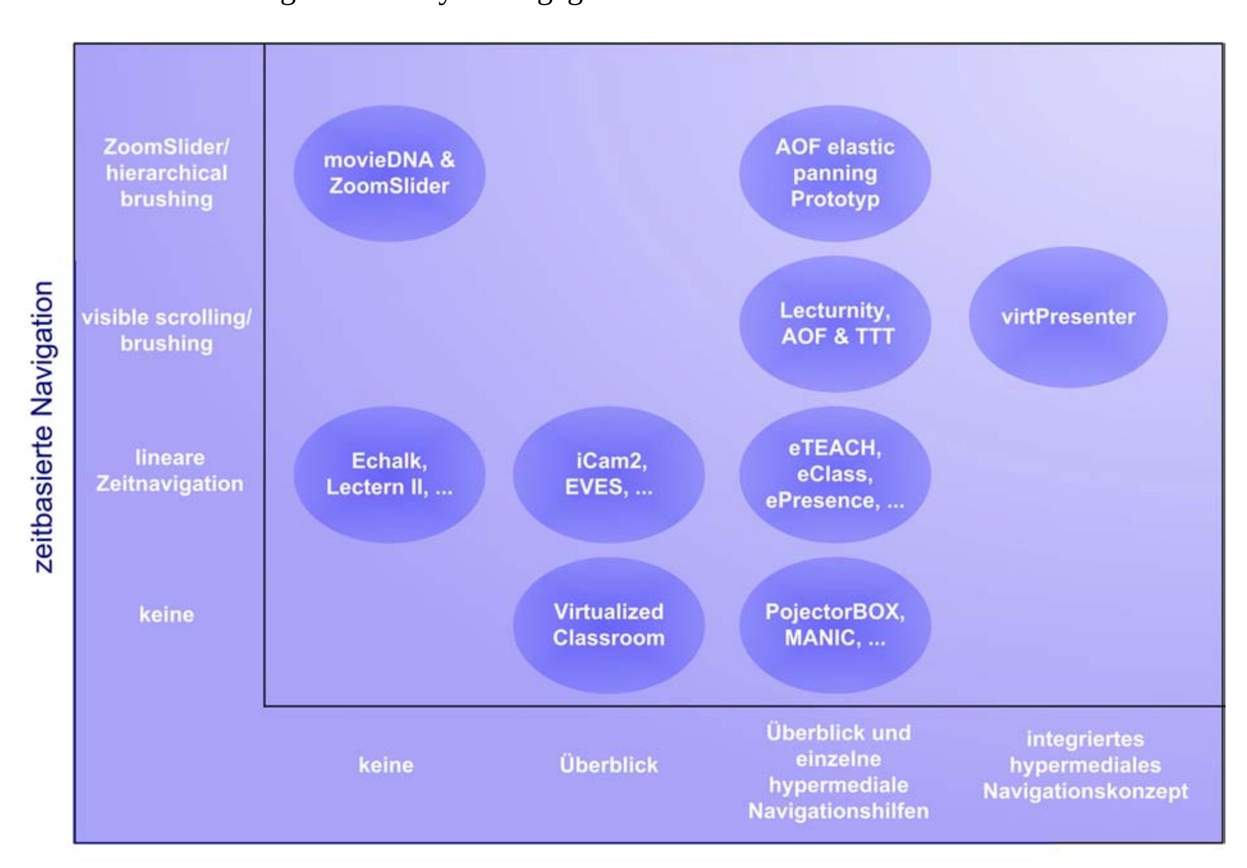



**Abb. 5-14: Zeit- und strukturbasierte Navigation in Vorlesungsaufzeichnungen**

Eine kompaktere Diskussion dieses Überblickes ist in [Mertens, Brusilovsky, Ishchenko & Vornberger 2006] zu finden.
Die Abbildung ist entlang der Achsen anhand der Fortschrittlichkeit der verwendeten Navigationsansätze geordnet. Wie in der Abbildung dargestellt, werden in vielen Systemen zeit- und strukturbasierte Ansätze kombiniert. Die in 5.1 bis 5.6 beschriebenen Systeme sind in der dritten Spalte angeordnet.
Es existieren nur sehr wenige Systeme, die über keine zeitbasierte Navigation verfügen. Hier sind nur der Virtualized Classroom [Zhu, McKittrick & Li 2004], die in 5.1 beschriebene ursprüngliche Variante des MANIC Interfaces und das System ProjektorBOX [Denoue, Hilbert, Adcock, Billsus & Cooper 2005] zu nennen. Beispiele für einfache zeitbasierte Navigation sind in sehr vielen Systemen wie E-Kreide [Knipping 2005], Lectern II [Joukov & Chiueh 2003], iCam2 [Zhang, Rui, Crawford, & He 2005], eTEACH (siehe 5.6), einigen Varianten von eClass [Brotherton 2001] oder ePresence (vgl. 5.4) zu finden. In diesen Ansätzen kann mittels eines Sliderinterfaces im Video navigiert werden. Weiterentwickelte zeitbasierte Navigationsansätze *wie random visible scrolling* (vgl. 4.2.1.2) oder das Zoom-Slider Interface [Hürst & Jarvers 2005] und *hierarchical brushing* (vgl. 4.2.1.2) sind nur wenig verbreitet.

| **Systeme** | **Hypermediale Navigationseigenschaften** | | | | |
|---|---|---|---|---|---|
| | Backtracking | Bookmarks | Footprints | Strukturelle Elemente | Volltext-suche |
| AOF jSync | - | - | - | ✓ | ✓ |
| MANIC | ✓ | ✓ | - | ✓ | ✓ |
| eClass/ Classroom 2000 | - | - | - | ✓ | ✓ |
| TTT | - | - | - | ✓ | ✓ |
| ePresence | - | - | - | ✓ | ✓ |
| eTEACH | - | - | ✓ | ✓ | - |
| Articulate Presenter | - | ✓ | - | ✓ | ✓ |
| VOD Clipping Tool | - | ✓ | - | - | - |

**Tabelle 5-7: Verteilung hypermedialer Navigationseigenschaften**

Die Verteilung strukturbasierter Navigationsansätze erstreckt sich von Systemen, die über keine strukturbasierte Navigation verfügen über eine Vielzahl von Systemen, die nur über eine Folienüberblicksfunktion verfügen bis hin zu Systemen, die einen Folienüberblick und einzelne hypermediale Navigationsfunktionen bieten. Ein integriertes hypermediales Navigationskonzept ist nur bei dem im Rahmen der vorliegenden Arbeit implementierten System virtPresenter vorhanden.
Das Vorhandensein vieler Systeme, die über keine strukturbasierten Navigationshilfen verfügen erklärt sich bei movieDNA und dem ZoomSlider aus dem Forschungsstatus der Systeme. Bei Systemen wie E-Kreide oder Lectern II ist das Fehlen strukturbasierter Navigationshilfen dadurch bedingt, dass die Systeme Tafelanschrieb aufzeichnen und daher keine Information über Folienwechsel oder Ähnliches gewinnen können. Systeme, die auf Basis von Bildschirmaufzeichnungen arbeiten sind prinzipiell von der gleichen Problematik betroffen. In einigen Ansätzen wie dem TTT (vgl. 5.3) und der Bildschirmaufzeichnungs-

variante von MANIC (vgl. 5.1) werden jedoch auch zu diesen Problemen Lösungsansätze entwickelt.
Für die in 4.3 verfolgte Argumentation ist die Tatsache interessant, dass viele Systeme über einzelne hypermediale Navigationsfunktionen verfügen und dass diese Navigationsfunktionen auch mit weiterentwickelten zeitbasierten Navigationsansätzen kombiniert werden. Diese Beobachtung spricht für die Wichtigkeit hypermedialer Navigationsfunktionen. Tabelle 5-7 zeigt eine Aufstellung zur Verteilung der einzelnen Navigationsfunktionen in den in 5.1 bis 5.6 betrachteten Systemen und einigen weiteren Systemen.
Der Arcticulate Presenter ist ein kommerzielles System, das über die Möglichkeit zum Setzen von Bookmarks, Volltextsuche und einen Folienüberblick verfügt [36]. Bookmarks können allerdings nicht zu beliebigen Zeitpunkten gesetzt werden sondern nur zu Beginn einer Folie. Das VOD Clipping Tool ist als Zusatzwerkzeug zu streamingbasierten Aufzeichnungen konzipiert [Miyahara, Kaiya & Kaijiri 2002]. Es ermöglicht, beliebige Ausschnitte von streamingbasierten Aufzeichnungen anzufertigen und zu verwalten und dient damit als externes Bookmarking-Werkzeug.

### 5.7.2 Alternative Navigationsansätze

Neben den bisher in dieser Arbeit behandelten Navigationsansätzen existieren noch einige weitere. In diesem Abschnitt werden diese Ansätze der Vollständigkeit halber kurz dargestellt.
Ein relativ nahe liegender Gedanke ist die **Nutzung von Transkripten der Sprachspur** der Aufzeichnung zur Volltextsuche im gesprochenen Text der Aufzeichnung. In einigen wenigen Systemen wie AOF [Hürst 2003b], dem BMRC Lecture Browser [Rowe, Harley, Pletcher & Lawrence 2001] und eClass [Brotherton 2001] wurde zu diesem Zweck bereits mit Spracherkennungssoftware experimentiert. Auch der E-Learning Media Navigator verwendet ein Transkript zur Volltextsuche [Dorai, Kermani & Stewart 2001]. In [Hürst 2004] werden einige Interfaces zur Volltextsuche in derartig erstellten Transkripten evaluiert. In [Munteanu, Baecker, Penn, Toms & James 2006] werden Ergebnisse einer Studie zur Nutzbarkeit von Transkripten unterschiedlicher Erkennungsqualität im Kontext von ePresence diskutiert.
Ansätze zur **Manipulation der Audio- und Videospur** beinhalten das Herausfiltern und Ausblenden von Sprechpausen wie in [Li, Gupta, Sanocki, He & Rui 2000] beschrieben. Weitere Ansätze erlauben das beschleunigte Abspielen der Aufzeichnung [He, Grudin & Gupta 2000] oder das Überspringen kleinerer Zeitintervalle wie in eTeach realisiert [Moses, Litzkow, Foertsch & Strikwerda 2002]. In [Hürst, Lauer & Götz 2004] wird ein spezielles Interface zum beschleunigten und zeitverzögerten Abspielen von Audiodateien beschrieben. Der Ansatz ist mit dem in 5.2.2 beschriebenen *elastic panning* zum Scrollen in Videodateien verwandt und wurde wie auch *elastic panning* im Kontext von AOF entwickelt.
In [Haubold & Kender 2003] wird der **Fokus der Aufnahme** als Navigationshilfe verwendet. Für verschiedene Foki existieren im Interface eigene Symbole wie Tafel, Publikum, Podium usw. die jeweils über Keyframes zu einzelnen Passagen angeordnet sind. Damit können Nutzer direkt Passagen ansteuern, in denen ein bestimmtes Präsentationsmedium verwendet wird oder eine Diskussion mit dem Publikum stattfindet. Die Kamerasteuerung wird in dem beschriebenen Ansatz von einem Kameramann vorgenommen. Die Zuordnung zu speziellen Szenen ist durch ein automatisiertes Verfahren zur Auswertung der Aufzeichnung realisiert.
[Altman, Chen & Low 2002] stellen einen Ansatz vor, in dem Vorlesungsaufzeichnungen nach **pädagogischen Kategorien** geordnet in einem hyperbolischen Baum navigiert werden können. Kategorien in diesem Ansatz sind beispielsweise Thema, Beispiel, Diskussion, Theorem und Diagramm. Zur Einordnung von Passagen der Aufzeichnung in die entspre-

chenden Kategorien müssen die jeweils verwendeten Folien manuell mit einem Schlüsselwort zur jeweiligen Kategorie zugeordnet werden.

### 5.7.3 Fazit

Grundzüge hypermedialer Navigation sind in sehr vielen Vorlesungsaufzeichnungssystemen vorhanden. Dabei sind alle der in 4.5.1 dargestellten hypermedialen Navigationseigenschaften mindestens in einem System vertreten. Dies stellt ein deutliches Indiz für die Eignung hypermedialer Navigation für Vorlesungsaufzeichnungen dar. Es ist allerdings auch anzumerken, dass mit Ausnahme des im Rahmen der vorliegenden Arbeit erstellten virtPresenter kein System mit einem vollständigen hypermedialen Navigationskonzept realisiert worden ist.

Neben hypermedialer Navigation sind auch eine Reihe weiterer Navigationsansätze in verschiedenen Forschungsprojekten entwickelt worden. Diese Ansätze bieten jedoch keine umfassenden Navigationskonzepte sondern lösen in der Mehrzahl Detailprobleme und können, wie beispielsweise die Transkripte der Sprachspur einer Aufzeichnung, ohne konzeptuelle Probleme in hypermediale Navigationsansätze integriert werden.

# 6 Ein hypermediales Navigationskonzept für Vorlesungsaufzeichnungen

In Kapitel 4 wurde ein Katalog von Anforderungen an die Umsetzung eines hypermedialen Navigationskonzeptes aufgestellt. In diesem Kapitel wird ein Ansatz erarbeitet, der diese Anforderungen erfüllt. Darauf aufbauend wird diskutiert, welche weiterführenden Navigationsansätze aus dem Forschungsgebiet Hypermedien auf den Anwendungsfall Vorlesungsaufzeichnungen sinnvoll übertragbar sind. Den Ergebnissen dieser Diskussion folgend wird daran anschließend ein Konzept zur sozialen Navigation in Vorlesungsaufzeichnungen vorgestellt. Das erarbeitete Konzept zur sozialen Navigation ist dabei vollständig in das in diesem Kapitel vorgestellte hypermediale Navigationskonzept für Vorlesungsaufzeichnungen integriert.

Dieses Kapitel dient in erster Linie der Erarbeitung und Diskussion der Funktionalität der Wiedergabekomponente auf einer konzeptuellen Ebene. Um eine Grundlage zur Umsetzung der in 4.5.1 beschriebenen Anforderungen an die jeweiligen Navigationselemente zu schaffen, wird zunächst die Realisierung der in 4.5.3 dargestellten Voraussetzungen geschildert. Diese umgekehrte Reihenfolge der Darstellung wurde gewählt, da die Art und Weise, in der die direkten Anforderungen an einzelne hypermediale Navigationselemente umgesetzt werden können, in hohem Maße von der zugrunde liegenden konzeptuellen Infrastruktur abhängt.

Einige der in diesem Abschnitt erarbeiteten Konzepte und Realisierungsansätze wurden in kürzerer Form bereits in [Mertens, Schneider, Müller & Vornberger 2004] und [Mertens, Brusilovsky, Ishchenko & Vornberger 2006] veröffentlicht.

## 6.1 Voraussetzungen

Die in 4.5.3 hergeleiteten Voraussetzungen für die Realisierung von hypermedialer Navigation in Vorlesungsaufzeichnungen umfassen die Visualisierung zeitlicher und struktureller Kontexte, die Skalierbarkeit verschiedener Elemente des Interfaces, die Verbindung von zeit- und strukturbasierter Navigation sowie die Animation von Folienschritten. Die in diesem Kapitel vorgestellten Lösungen basieren in Teilen auf den in den Kapiteln 4 und 5 dargestellten Beobachtungen.

Im Folgenden werden zunächst Ansätze zur Visualisierung zeitlicher und struktureller Kontexte in Vorlesungsaufzeichnungen entwickelt. Die Ansätze basieren auf den beiden Orientierungshilfen, die einen globalen Überblick über die Aufzeichnung geben. Dies sind der Folienüberblick und die Zeitleiste. Beide Orientierungshilfen sind in vielen Wiedergabeinterfaces zu finden und bieten in der Regel auch eine Einordnung der aktuellen Wiedergabeposition in Gesamtzeit und Struktur der Aufzeichnung (vgl. Abb. 6-1).

In 6.1.3 wird aufbauend auf den in 6.1.1 in 6.1.2 dargestellten Ansätzen ein Konzept zur bidirektionalen Verbindung von zeit- und strukturbasierter Navigation in Vorlesungsaufzeichnungen entwickelt.

Die Realisierung der in Abschnitten 4.5.3.1 bis 4.5.3.5 erörterten Anforderungen der Skalierbarkeit von verschiedenen Elementen des Interfaces sowie der Animation von Folienschritten wird aufgrund der eher technischen Natur der damit verbundenen Fragestellungen zusammenfassend in 6.1.4 behandelt.

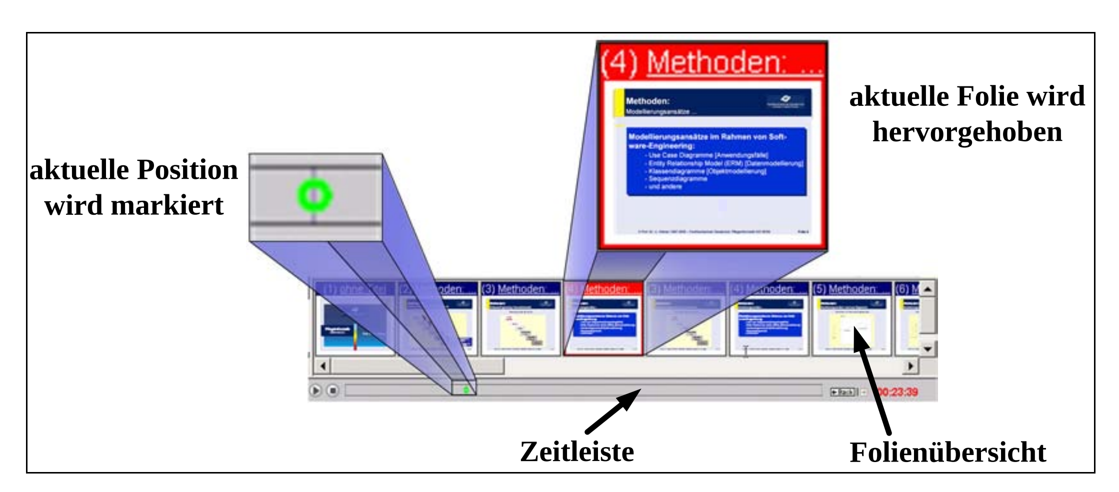



**Abb. 6-1: Zeitleiste und Folienüberblick als globale Orientierungshilfen im virtPresenter**

### 6.1.1 Visualisierung zeitlicher Kontexte

Zur Visualisierung zeitlicher Kontexte muss zunächst ein zeitbasiertes Überblickselement im Benutzerinterface vorhanden sein. Wie viele andere Vorlesungsaufzeichnungssysteme auch, verfügt das im Rahmen dieser Arbeit entwickelte Wiedergabeinterface daher über eine zur Navigation nutzbare Zeitleiste.
Beispiele für die Nutzung derartiger zeitbasierter Überblickskomponenten zur Visualisierung zeitlicher Kontexte in Vorlesungsaufzeichnungen sind die *interactive timeline* des ePresence Systems 5.4 und die Markierung von Darstellungsdauern von Folien durch vertikale Striche im BMRC Lecture Browser, der Wiedergabekomponente des PRISM Toolkits und dem Cornell Lecture Browser (vgl. 4.2.2.1). [Li, Gupta, Sanocki, He & Rui 2000] stellen einen Video Browser vor, in dem derartige Markierungen in verschiedenen Farben zur Visualisierung verschiedener Typen von Unterteilungen wie Folienwechsel oder Notizen genutzt werden können.

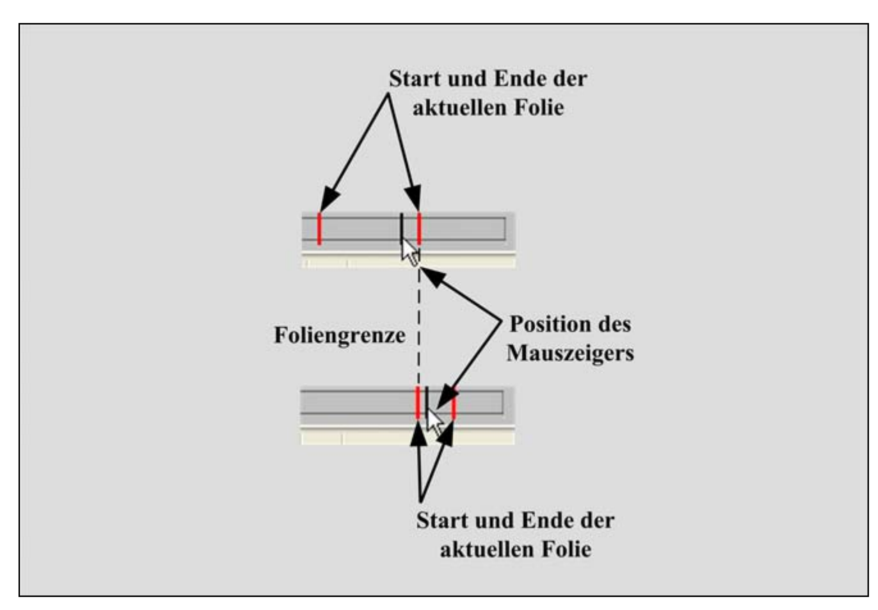



**Abb. 6-2: Im virtPresenter werden bei der Navigation in der Zeitleiste die Grenzen der jeweils aktuellen Folie hervorgehoben**

In den voran genannten Ansätzen werden alle Markierungen permanent angezeigt. Diese Darstellung kann allerdings bei feingranularer Segmentierung leicht unübersichtlich werden.

[65] Dies stellt für die angestrebte Verbindung von Zeit und Struktur ein Problem dar, da hierfür eine Hervorhebung einzelner Markierungen notwendig ist. So muss in der Zeitleiste beispielsweise die aktuelle Folie hervorgehoben werden können (vgl. 6.1.3). Das im Rahmen dieser Arbeit entwickelte Navigationsinterface basiert daher auf einer kontextsensitiven Visualisierung zeitlicher Kontexte wie in Abb. 6-2 dargestellt.
Wird auf der Zeitleiste eine beliebige Position mit einem Mausklick ausgewählt, werden die zugehörigen Foliengrenzen durch senkrechte, rote Markierungen hervorgehoben. Verändert der Nutzer die Position des Mauszeigers so, dass der markierte Zeitpunkt außerhalb der aktuell markierten Foliengrenzen liegt, werden die Markierungen aktualisiert, um die Foliengrenzen der neuen Position anzuzeigen.

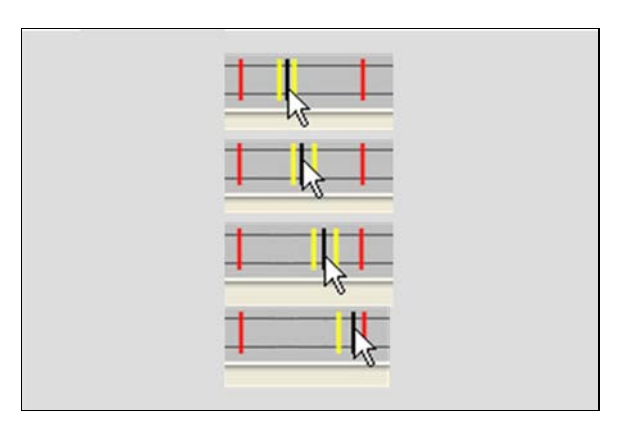


**Abb. 6-3: Markierung von Folienanimationen bei zeitbasierter Navigation im virtPresenter**

Grenzt die aktuelle Position an eine Folienanimation, werden die angrenzenden Start- und Endzeitpunkt der umgebenden Animation durch gelbe vertikale Linien hervorgehoben (siehe Abb. 6-3). Befindet sich die Position zwischen Folienanfang und erster Animation oder letzter Animation und Folienende, wird jeweils nur der angrenzende Start- bzw. Endpunkt markiert.
Auf diese Weise wird die jeweilige Abspielposition nicht nur in den zeitlichen Gesamtverlauf der Aufzeichnung eingebettet, sondern kann auch zu lokalen Ereignissen wie Folienwechseln oder Animationen in Bezug gesetzt werden. Darüber hinaus kann diese Form der Markierung zeitlicher Grenzen auch aus strukturbasierten Navigationshilfen heraus zu deren zeitlicher Einordnung genutzt werden (vgl. 6.1.3 ).

### 6.1.2 Visualisierung struktureller Kontexte

Zur Visualisierung der Vortragsstruktur wird, wie bereits in 6.1 erwähnt, in vielen Fällen eine Leiste mit miniaturisierten Abbildungen der verwendeten Folien verwendet. Die Folien werden dabei in der Reihenfolge angezeigt, in der sie während der Aufzeichnung präsentiert worden sind. Ein farbliches Hervorheben von Folien in dieser Übersichtleiste kann nicht nur wie in Abb. 6-1 dargestellt zur Markierung der Position der aktuell in der Wiedergabe gezeigten Folie im strukturellen Gesamtkontext der Aufzeichnung genutzt werden. Diese Technik kann auch zur Anzeige der Sprungziele verschiedener Navigationshilfen wie beispielsweise Bookmarks (vgl. 6.2.3) dienen. Im virtPresenter sind alle Navigationshilfen mit Ausnahme der Zeitleiste maussensitiv und markieren bei Überfahren mit dem Mauszeiger die entsprechende Zielfolie in der Folienübersicht (siehe Abb. 6-4). In der Zeitleiste ist diese Funktion nur bei gedrückter rechter Maustaste aktiviert (vgl. 6.1.3). Die

[65] Vgl. Abb. 5-8 auf Seite 98

Navigationselemente bieten damit eine Vorschaufunktion zum Ziel der verknüpften Navigationsaktion. Im Bereich klassischer text- und bildbasierter Hypermedien bietet das System Hyperscout eine ähnliche Funktionalität [Weinreich, Obendorf & Lamersdorf 2004]. Dort wird beim Überfahren eines Links eine Kurzzusammenfassung der verlinkten Webseite angezeigt. Im Fall folienbasierter Präsentationen stellt die Folie selbst eine solche Kurzzusammenfassung dar.
In der Regel beschreibt der zu einem Zeitpunkt angezeigte Animationsschritt den umgebenden Zeitabschnitt besser als der in der Übersicht standardmäßig angezeigte Endzustand der jeweiligen Folie. Um diesem Umstand gerecht zu werden, wird bei Überfahren einer Navigationshilfe, die mit einem Animationsschritt verbunden ist, das Aussehen der Folie direkt nach dem betreffenden Animationsschritt angezeigt. So können in vielen Fällen Rückschlüsse auf den thematischen Fokus des verlinkten Zeitabschnittes gezogen werden. Ein weiterer Vorteil der Anzeige des konkreten Folienzustandes im Gegensatz zu einer standardisierten Folienendansicht zeigt sich, wenn die Folie Animationsschritte enthält, die Objekte auf der Folie verschieben oder verschwinden lassen. Derartig animierte Objekte sind in der Folienendansicht nicht oder anders dargestellt als in der konkreten Foliendarstellung zum jeweiligen Zeitpunkt. Wie in Abb. 6-4 deutlich wird, kann durch dynamische Darstellung des konkreten Folienzustandes daher deutlich mehr Information vermittelt werden als durch die statische Darstellung des Endzustandes der Folie.

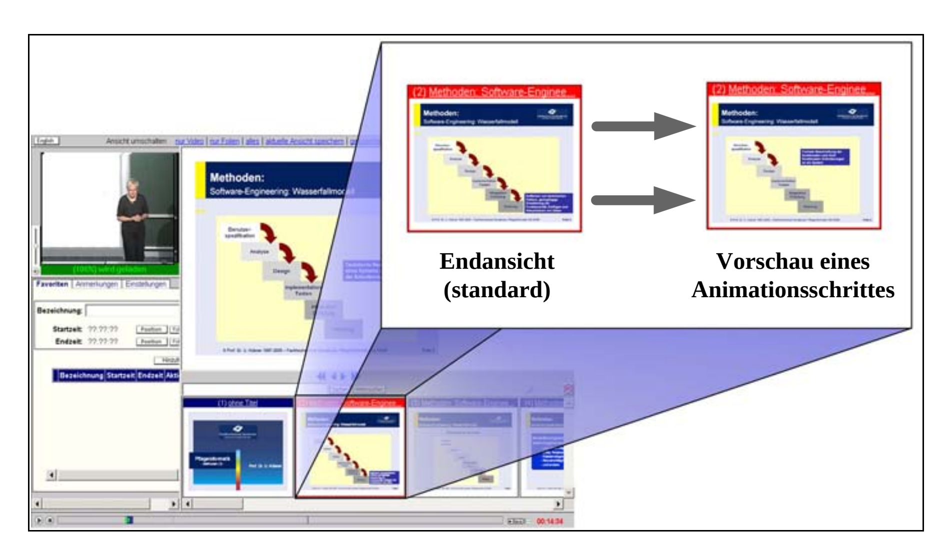



**Abb. 6-4: Vorschau der Ansprungpunkte von Navigationshilfen in der Folienübersichtsleiste des virtPresenters. Animierte Folien werden in der Vorschau mit der Ansicht der zum Navigationszielpunkt aktuellen Animation dargestellt.**

Durch die Anzeige von Sprungzielen anderer Navigationshilfen geht die Erweiterung des Nutzungsspektrums der Folienübersicht deutlich über die Funktionalität hinaus, die in anderen Systemen zur Vorlesungsaufzeichnung zu finden ist. Teilweise ist diese Technik verwand mit visible Scrolling oder Brushing (vgl. 4.2.1.2), in diesen Ansätzen wird sie jedoch nur in Verbindung mit zeitleistenbasierter Navigation eingesetzt.

### 6.1.3 Verbindung von Zeit und Struktur

Das im Rahmen der vorliegenden Arbeit implementierte Konzept zur Verbindung von zeit- und strukturbasierter Navigation basiert wie bereits erwähnt auf den in den beiden vorangegangenen Teilabschnitten beschriebenen Überblickskomponenten. Im Gegensatz zu den in anderen Systemen zur Wiedergabe von Vorlesungsaufzeichnungen realisierten Ansätzen ist

die hier vorgestellte Verbindung zwischen Zeit und Struktur nicht unidirektional nur von Zeit zu Struktur oder von Struktur zu Zeit realisiert, sondern bidirektional.
So ist das Überfahren der Zeitleiste mit dem Mauszeiger bei gedrückter rechter Maustaste an einen an das in [Ponceleon & Dieberger 2001] beschriebene Brushing angelehnten Ansatz gekoppelt (vgl. dazu auch 4.2.1.2). Bei dem im Rahmen der vorliegenden Arbeit realisierten Ansatz wird jeweils die Folie in der Folienüberblickskomponente farblich hervorgehoben, die dem gewählten Zeitpunkt zugeordnet ist. Zudem wird auch bei der Navigation über die Zeitleiste die in 6.1.2 geschilderte Darstellung des jeweils aktuellen Animationsschrittes genutzt (siehe Abb. 6-5).

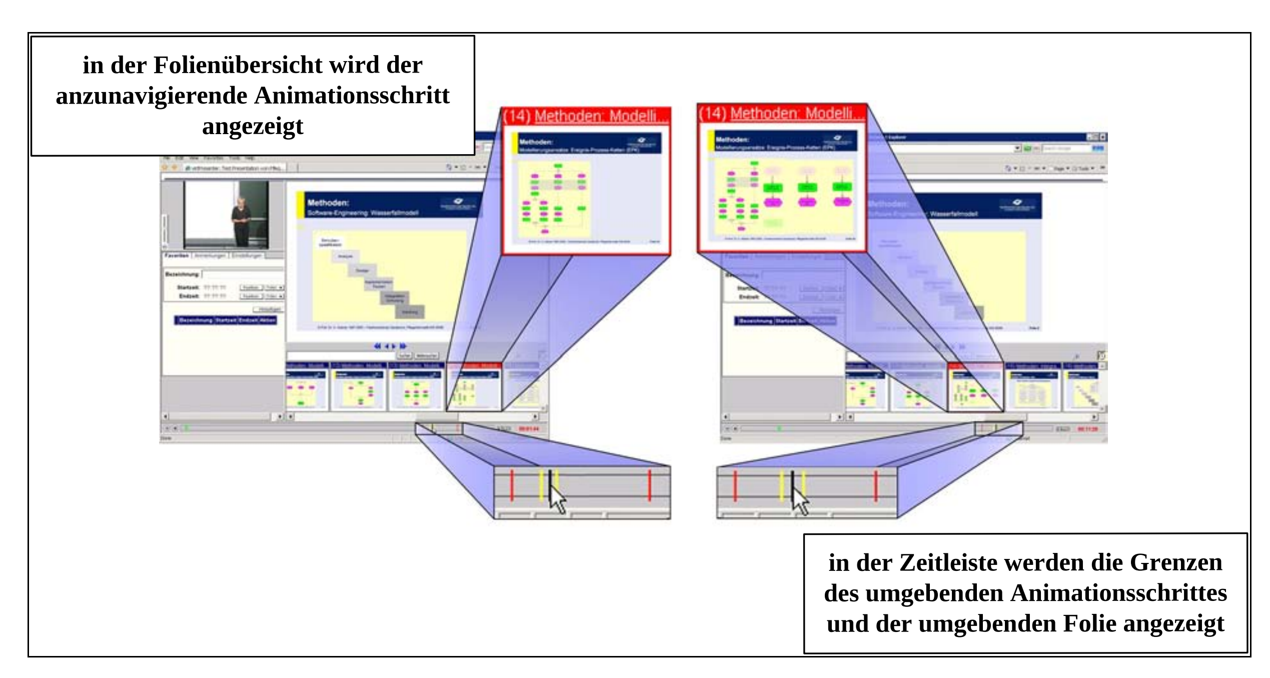



**Abb. 6-5: Verbindung von zeit- und strukturbasierter Navigation im virtPresenter. Die aktuelle Position wird sowohl in der Zeitleiste als auch im Folienüberblick markiert.**

Da die Grenzen der aktuellen Folie und ggf. des aktuellen Animationsschrittes auf der Zeitleiste markiert sind, kann die Vorschaufunktion in der Folienübersichtsleiste nicht nur zur thematischen Einordnung des auf der Zeitleiste markierten Zeitpunktes, sondern des gesamten zugehörigen Zeitintervalls genutzt werden. Umgekehrt liefert die Länge des Zeitintervalls Hinweise darauf, wie intensiv der auf der Folie dargestellte Sachverhalt behandelt wurde. Diese Information ist insbesondere bei mehrmaligem Zeigen einer Folie oder eines Animationsschrittes während der Aufzeichnung hilfreich.
Die in Abb. 6-5 dargestellte synchrone Visualisierung des zeitlichen und des strukturellen Kontextes eines Navigationsziels ist nicht nur für die Zeitleiste sondern für alle im weiteren Verlauf dargestellten Navigationshilfen implementiert. So werden auch beim Überfahren strukturbasierter Navigationselemente die entsprechenden Markierungen in der Zeitleiste angezeigt. Wie im weiteren Verlauf dieses Kapitels deutlich wird, zieht sich das Konzept der gleichzeitigen Markierung struktureller und zeitlicher Kontexte konsequent durch alle im Rahmen dieser Arbeit implementierten Navigationshilfen und stellt damit eine bidirektionale Verbindung zwischen zeit- und strukturbasierter Navigation her.

### 6.1.4 Weitere Anforderungen

Der in 4.5.3.1 gestellten Forderung nach einer freien Skalierbarkeit verschiedener Elemente des Interfaces wird beim virtPresenter durch einen framebasierten Ansatz nachgekommen. Im Gegensatz zu fensterbasierten Ansätzen kommt es bei framebasierten Ansätzen nicht zur Überdeckung einzelner Bestandteile des Interfaces und das Interface kann als ein kohärentes

Objekt wahrgenommen werden [Mertens, Friedland & Krüger 2006], wodurch generell ein besserer Überblick realisiert werden kann.
Die im Zusammenhang mit der Skalierung des Folienüberblicks auftretenden Problematik der ungenauen Darstellung von Folieninhalten [Lauer & Trahasch 2005] wird durch den Einsatz des Vektorgrafikformates SVG vermieden (vgl. 7.4.1.1).
Die in 4.5.3.5 erörterte Forderung nach der Wiedergabe von einzelnen Animationsschritten einer Folie wird durch die in 8.1 behandelte Zerlegung einer Folie in ihre Animationsschritte nachgekommen. Damit kann jeder Animationsschritt als unabhängiges Bild betrachtet und so beispielsweise auch zur Darstellung in der Vorschauansicht im Folienüberblick (vgl. 6.1.2) genutzt werden.

## 6.2 Realisierung des Navigationskonzeptes

Wie bereits in Kapitel 4 erarbeitet wurde, sollte ein hypermediales Navigationskonzept für Vorlesungsaufzeichnungen die Navigationshilfen Volltextsuche, Backtracking, Footprints, strukturelle Elemente und Bookmarks enthalten. Dieser Abschnitt behandelt die Umsetzung der zugrunde liegenden Konzepte für den Anwendungsfall Vorlesungsaufzeichnungen und die Einpassung der modifizierten Navigationshilfen in das Wiedergabeinterface des virtPresenter. Ein kurzer Überblick zur räumlichen Anordnung der Navigationshilfen im Wiedergabeinterface wird in Abb. 6-6 gegeben.

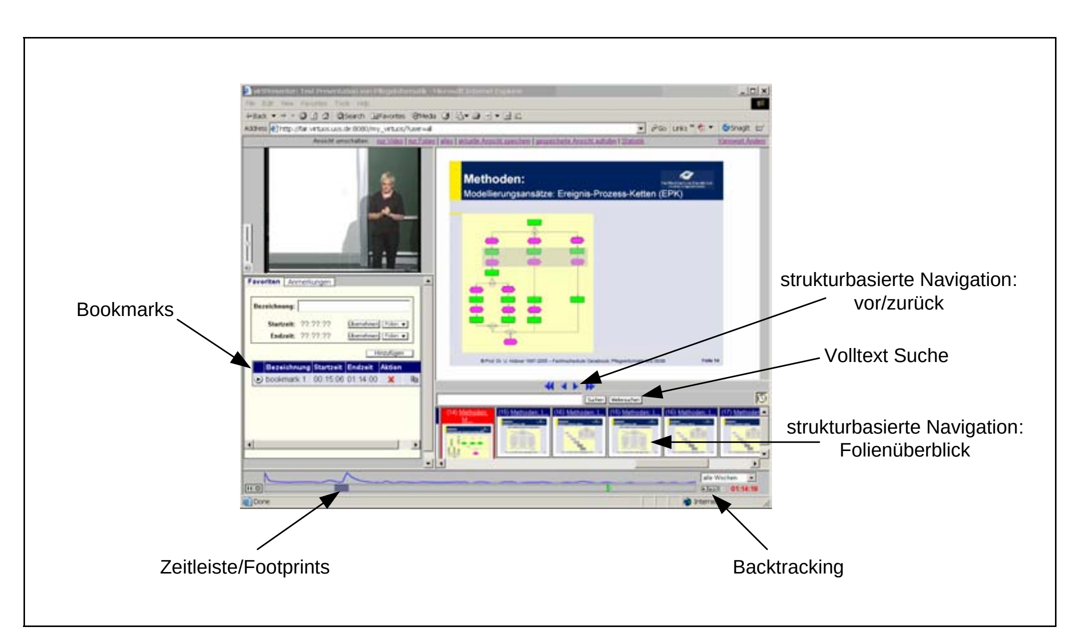



**Abb. 6-6: Anordnung der hypermedialen Navigationshilfen im Wiedergabeinterface**

Die Betrachtung der Arbeitsweise hypermedialer Navigation in Kapitel 4 hat gezeigt, dass die hypermedialen Navigationshilfen nicht unabhängig voneinander eingesetzt werden. Vielmehr ist gerade das Zusammenspiel aller Navigationselemente für eine effektive Navigationsunterstützung essentiell (vgl. 4.5.2). Um einen Bezug zwischen den einzelnen Navigationshilfen herzustellen, sind daher alle Navigationshilfen sowohl an die Zeitleiste als auch den Folienüberblick gekoppelt. Diese Kopplung ist bei den einzelnen Navigationselementen abhängig von deren Integration in das Wiedergabeinterface unterschiedlich realisiert. So sind Footprints direkt in die Zeitleiste integriert und einige strukturelle Elemente in den Folienüberblick. Wie genau die Verbindung für die jeweiligen Navigationshilfen gestaltet ist, wird im weiteren Verlauf dieses Abschnittes deutlich.
Auf eine ausführliche Darstellung des allgemeinen Nutzens der einzelnen Navigationselemente wird in diesem Abschnitt verzichtet, da diese bereits in 4.4.2 zu finden ist. Lediglich

implementationsspezifische Erweiterungen werden bezüglich des durch sie ermöglichten Zusatznutzens diskutiert.

### 6.2.1 Footprints

Footprints sind im virtPresenter zeitbasiert realisiert. Dadurch wird die Problematik mehrdeutiger Aussagen von Footprints umgangen, die sich bei einem strukturbasiertem Ansatz zur Umsetzung dieser Navigationshilfe im zeitbasierten Medium Vorlesungsaufzeichnung stellt (vgl. 4.5.1.3).
Um Platz bei der Gestaltung des Interfaces zu sparen und um Footprints enger an die zeitbasierte Navigation in der Aufzeichnung zu koppeln, werden Footprints direkt auf der Zeitleiste dargestellt. Wie in Abb. 6-7 gezeigt, ist die Markierung bereits besuchter Abschnitte der Aufzeichnung auf der Zeitleiste durch eine blaue Einfärbung realisiert. Ein ähnlicher Visualisierungsansatz wird von [Lauer & Busl 2007] zur Navigation in Audiodateien verwendet.

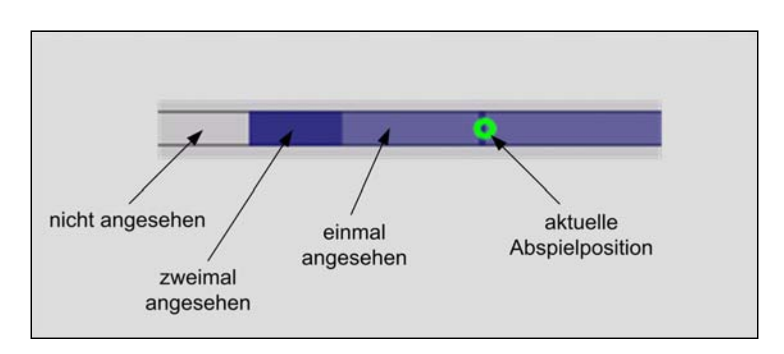



**Abb. 6-7: zeitbasierte Footprints im virtPresenter**

Durch die direkte Integration von Footprints in die Zeitleiste, können diese nicht nur annavigiert werden, sondern Nutzer haben auch die Möglichkeit, innerhalb von Footprints zu navigieren. Aufgrund der Kopplung von zeit- und strukturbasierter Navigation kann ein Footprint bei gedrückter rechter Maustaste abgefahren werden und in der Folienübersicht wird die zum jeweiligen Zeitpunkt des Footprints gezeigte Folienansicht dargestellt (vgl. 6.1.3). Durch die kontextsensitive Markierung von Foliengrenzen und Animationsschritten in der Zeitleiste können Footprints darüber hinaus auch zur Darstellungsdauer der jeweils gezeigten Folie oder des jeweils gezeigten Animationsschrittes in Relation gesetzt werden. So kann auch festgestellt werden, ob sich der Nutzer nur einen Teil des Vortrags zu einer Folie oder den gesamten zur Folie gehalten Vortrag angesehen hat. Eine semitransparente Darstellung der Footprint-Markierungen ermöglicht es dem Nutzer zu unterscheiden, ob ein Abschnitt ein, zwei, drei Mal oder öfter angeschaut wurde. So werden auch öfter angeschaute kurze Passagen in einem längeren Footprint erkennbar. Dies kann zum Beispiel dann hilfreich sein, wenn ein Nutzer zunächst eine längere Passage der Aufzeichnung angesehen hat und später nur gezielt schwierige Teile dieser Passage wiederholt hat (vgl. 3.6.1).[66]
Ein weiterer Vorteil der in Abb. 6-7 dargestellten Art der Integration von Footprints in die Zeitleiste ergibt sich aus der Tatsache, dass die Ansprungziele aller Navigationselemente in der Zeitleiste markiert werden (siehe 6.1.2). Damit sind Footprints von allen Navigationselementen aus nutzbar und reichern die jeweilige Navigationshilfe so um die Zusatzinformation an, ob die zugehörige Navigationsaktion in einen bereits bekannten Bereich der Aufzeichnung führt oder das Abspielen einer noch nicht angesehenen Passage bewirkt.

[66] Eine ähnliche Form der Markierung des Nutzungsgrades von Inhalten durch Farbintensität wird von [Brusilovsky, Chavan & Farzan 2004] im Kontext sozialer Navigation in klassischen Hypermedien vorgestellt.

### 6.2.2 Strukturelle Elemente

Der virtPresenter ermöglicht Navigation anhand zweier verschiedener Arten von strukturellen Elementen. Zum Einen kann mit Hilfe der in Abb. 6-8 dargestellten Bedienelemente direkt die zeitlich vorgelagerte bzw. nachfolgende Folie oder auch Animation angesteuert werden. Zum Anderen ist der Folienüberblick interaktiv gestaltet, wie weiter unten beschrieben.
Der Tatsache, dass nicht alle Folien Animationen enthalten und dass Folienanimationen zu Folienwechseln in Beziehung gesetzt werden müssen, wird durch einen speziellen Mechanismus begegnet. Ist eine Folie nicht animiert, sind die Pfeilsymbole zur Anwahl der vorangehenden und folgenden Animation durch Ausgrauen als deaktiviert gekennzeichnet. Liegt der aktuelle Zeitpunkt vor der ersten Animation der gerade gezeigten Folie, wird das Pfeilsymbol zur Anwahl der vorangehenden Animation deaktiviert. Wird die Folie bereits in der Endansicht angezeigt, wird das Symbol zur Anwahl des nachfolgenden Animationsschrittes deaktiviert.

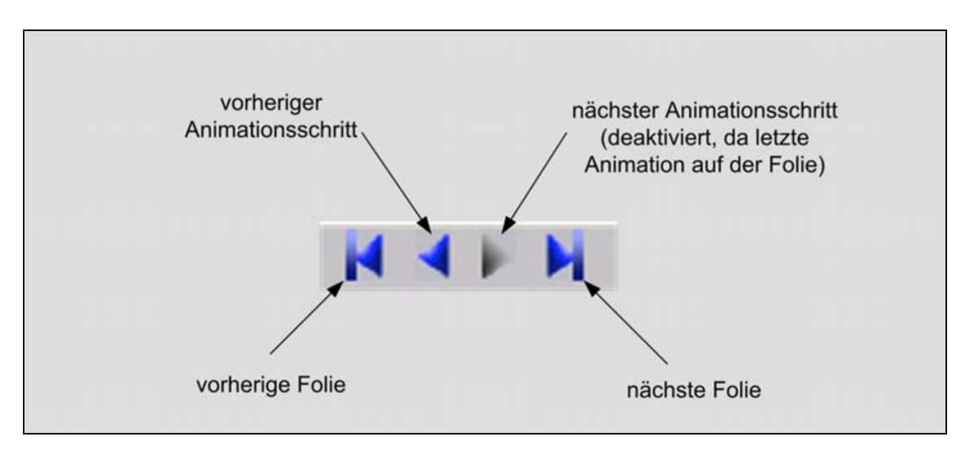



**Abb. 6-8: Navigationselemente zur Ansteuerung benachbarter Animationsschritte und Folien**

Beim Überfahren der inneren, zur Annavigation von Animationsschritten konzipierten Pfeilsymbole mit dem Mauszeiger werden die Grenzen der jeweiligen Animation und der umgebenden Folie in der Zeitleiste angezeigt, wie in Abb. 6-9 dargestellt. Beim Überfahren der äußeren, zur Annavigation von Folien genutzten Pfeilsymbole werden die Grenzen der Folie markiert.
Im Folienüberblick wird bei Überfahren der äußeren Pfeilsymbole mit dem Mauszeiger jeweils die entsprechende Folie durch eine rote Umrandung markiert. Beinhaltet die Folie Animationsschritte, wird sie in der Anfangsansicht dargestellt (vgl. Abb. 6-4 auf Seite 114). Wird der Mauszeiger von dem jeweiligen Pfeilsymbol weg bewegt, wechselt die Folienübersicht wieder zur Darstellung der Folie in der Endansicht. Bei Überfahren der inneren Pfeilsymbole wird die durch den jeweiligen Pfeil ansteuerbare Folie in der zum jeweiligen Animationsschritt aktuellen Ansicht dargestellt und ebenfalls durch eine rote Umrandung markiert.

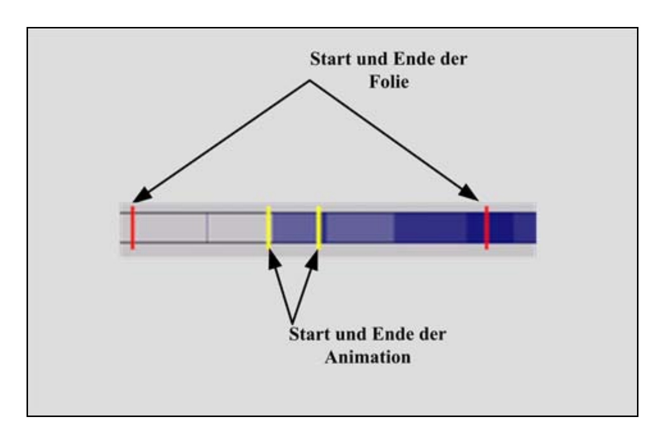



**Abb. 6-9: Markierung der Grenzen von Folie und Animation in der Zeitleiste**

Neben den Pfeilsymbolen zur Ansteuerung von zeitlich nahen Animationsschritten und Folien verfügt das Interface der Wiedergabekomponente noch über eine zweite Kategorie von strukturellen Elementen zur Navigation in der Aufzeichnung. Sowohl die Titel der angezeigten Folien als auch animierte Folienelemente, die in der Endansicht der Folie dargestellt werden, sind mit dem Zeitpunkt des Erscheinens der Folie bzw. des betreffenden Animationsschrittes als Ansprungpunkt in der Aufzeichnung verbunden.[67] Wie in Abb. 6-10 verdeutlicht, wird es damit möglich die jeweiligen Animationspunkte direkt aus der Übersichtsfolie heraus anzusteuern. Ein Mausklick auf ein animiertes Element der Folie genügt und der Abspielvorgang beginnt beim angesteuerten Animationsschritt.

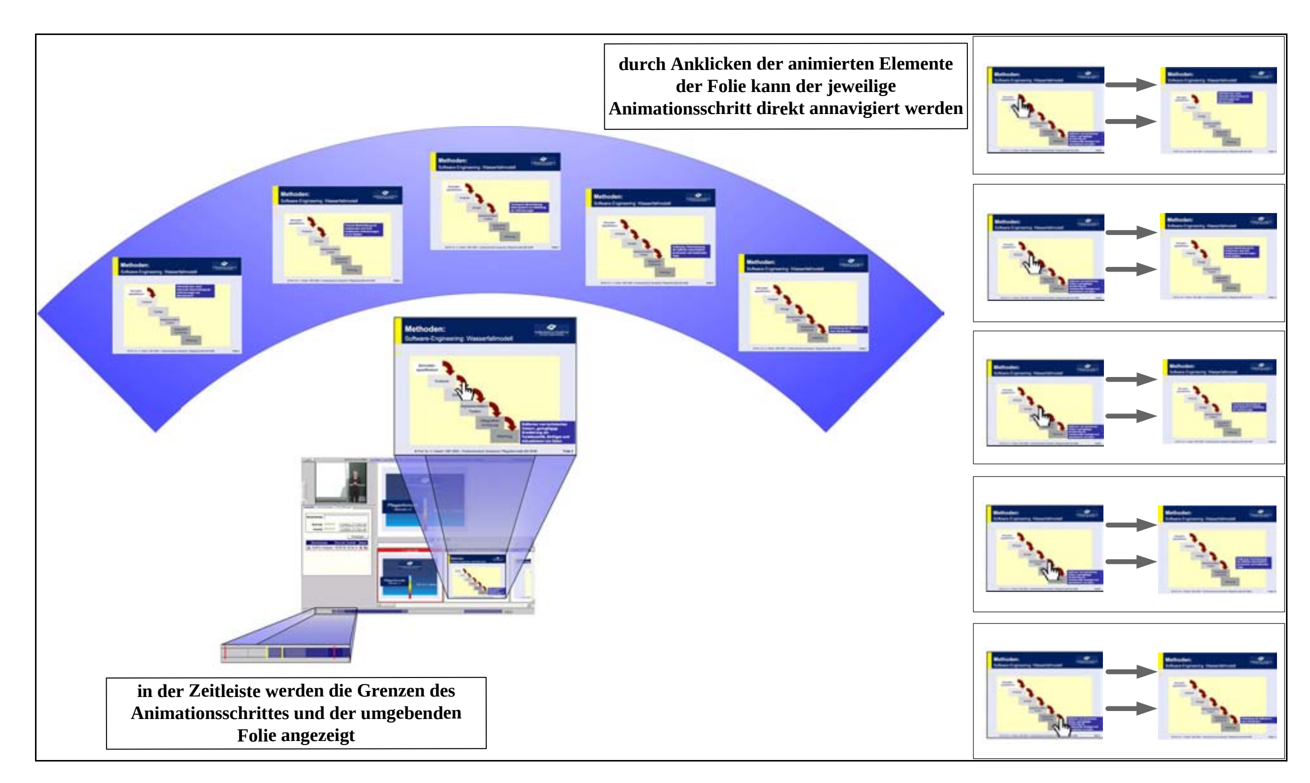



**Abb. 6-10: Verwendung der Folienübersichtsleiste im virtPresenter als strukturelles Navigationselement**

Um das Auffinden animierter Folienbestandteile zu erleichtern, verändert sich die Form des Mauszeigers beim Überfahren dieser Elemente. In der Zeitleiste werden die Grenzen der aktuellen Animation durch gelbe senkrechte Striche und die der umgebenden Folie durch rote senkrechte Striche markiert. Die Visualisierung des zeitlichen Kontextes folgt damit

[67] Ist ein Element mehrmals animiert, wird die erste Animation des Elementes als Ansprungpunkt verwendet.

dem in 6.1.1 erarbeiteten Ansatz (vgl. auch Abb. 6-9) und stellt so eine Verbindung zur Zeitachse her.

### 6.2.3 Bookmarks

Wie bereits in 4.5.1.5 erläutert, sind Bookmarks im virtPresenter als zeitbasierte Navigationshilfe implementiert. Sie können damit unabhängig von der Struktur des Vortrags gesetzt werden. So kann ein Bookmark mitten in einer Folie beginnen und auch mitten in einer Folie enden. Da Bookmarks prinzipiell einen alternativen Zugang zu einem hypermedialen Informationsuniversum darstellen und dieses auf eine andere Weise als die vorgegebene Struktur ordnen, sind sie nicht in eine der beiden Übersichtskomponenten integriert. Sowohl bei der Erstellung von Bookmarks als auch bei deren Nutzung sind jedoch Verbindungen zu zeit- und strukturbasierter Navigation realisiert.
Zur zeitbasierten Erstellung von Bookmarks kann die aktuelle Abspielposition per Knopfdruck als Start- oder Endpunkt des Bookmarks gesetzt werden. Alternativ können auch die Anfangspunkte der in der Aufzeichnung gesetzten Folien über eine Drop-Down Liste der Folientitel als Start- oder Endpunkt gesetzt werden. Beide Erstellungsformen können vom Nutzer kombiniert werden. Die Definition von Endpunkten ist optional. Wird kein Endpunkt definiert, spielt das Bookmark bis zum Ende der Aufzeichnung. Zur nutzerseitigen Ordnung der Bookmarks besteht die Möglichkeit, Bookmarks mit Titeln zu versehen.
Wie in Abb. 6-11 ersichtlich, ist die Einbindung der Visualisierung von Bookmarks in den zeitlichen Gesamtkontext der Aufzeichnung vergleichsweise einfach realisiert. Bei Überfahren des Play-Knopfes eines Bookmarks werden Anfang und ggf. Ende des Bookmarks durch zwei senkrechte rote Striche in der Zeitleiste markiert.

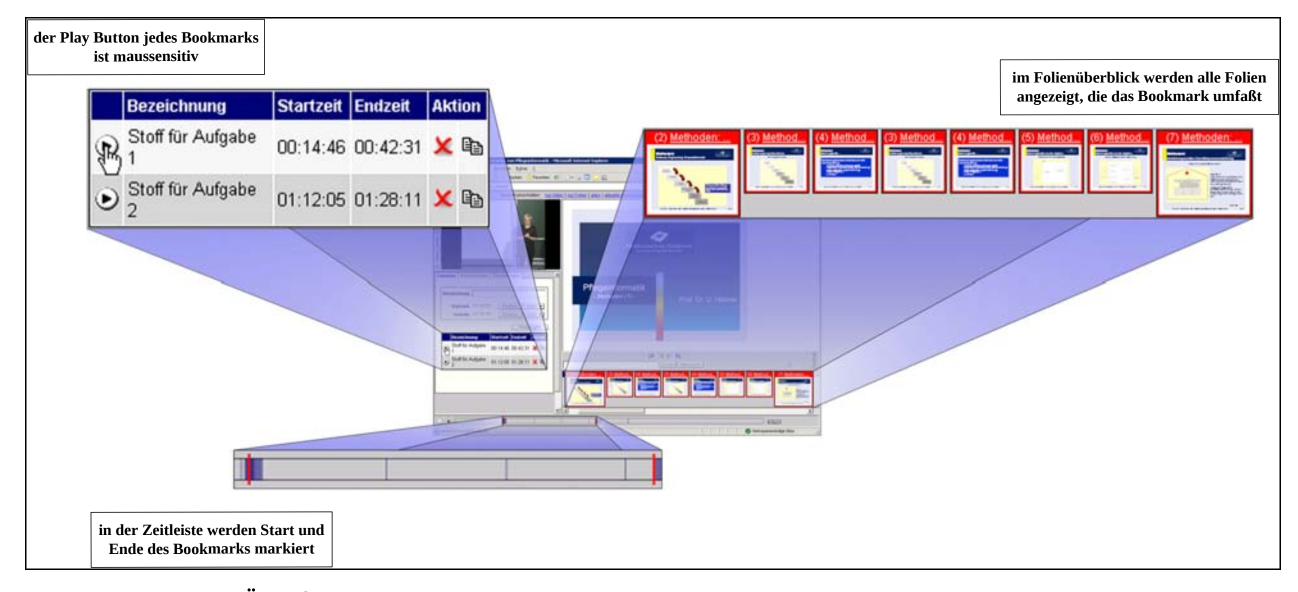



**Abb. 6-11: Beim Überfahren des Play Buttons eines Bookmarks werden die zugehörigen Folien sowie Start- und Endpunkt des Bookmarks in Zeitleiste und Folienüberblick markiert.**

Im Folienüberblick werden beim Überfahren des Play-Knopfes eines Bookmarks alle zugehörigen Folien gezeigt und durch eine rote Umrahmung markiert (siehe Abb. 6-11). End- und Startfolie des Bookmarks werden dabei in voller Größe und in der Ansicht dargestellt, in der die jeweilige Folie zu dem betreffenden Zeitpunkt angezeigt wird. Beginnt ein Bookmark also bei einem bestimmten Animationsschritt, wird der Zustand der Folie zu diesem Animationsschritt angezeigt. Die Folien zwischen Anfang und Ende des Bookmarks werden verkleinert dargestellt, um eine Anzeige aller zugehörigen Folien zu ermöglichen und so einen Überblick über das gesamte Bookmark zu gewährleisten.

Eine Erweiterung des klassischen Bookmarkkonzeptes stellt das Exportkonzept von Bookmarks im virtPresenter dar. Per Mausklick kann ein Bookmarks als herkömmlicher Link in die Zwischenablage kopiert werden und anschließend beispielsweise in einer Mail oder einem Online-Forum kommuniziert werden. Durch Anklicken des Links wird nach dem Laden der Präsentation automatisch das Bookmark abgespielt. So können beliebige Abschnitte der Aufzeichnung extern referenziert und genutzt werden. Wie in [Day, Foley, Groeneweg & Van der Mast 2005] postuliert, ergibt sich durch die daraus resultierende Möglichkeit, gezielt Fragen zu definierten Inhalten online zu diskutieren ein didaktischer Mehrwert.[68] Ein entfernt verwandter Ansatz wird in [Mu 2005] beschrieben. Das dort beschriebene System verfügt über ein Chat-Werkzeug, das in ein Interface zur Wiedergabe von Videodateien integriert ist. Dabei können Chat-Botschaften auf diskrete Zeit-Indizes im Video verweisen. Möglichkeiten zum Export solcher Verweise oder zur Erstellung nutzerdefinierter Passagen, die auch über einen Endpunkt verfügen, scheinen jedoch nicht gegeben zu sein. Prinzipiell implementiert das Exportkonzept die selbe Funktionalität, die in klassischen webbasierten Hypermedien durch das Kopieren einer Webadresse gegeben ist und stellt damit ein Äquivalent zu einer viel genutzten Eigenschaft textbasierter hypermedialer Webinhalte dar.

### 6.2.4 Volltextsuche

Die Volltextsuche umfasst den gesamten Text aller in der Folienübersicht angezeigten Folien. Wird ein Suchbegriff auf einer Folie gefunden, wird auf der jeweiligen Folie in der Folienübersichtsleiste der Textblock, in dem sich der Suchbegriff befindet, über eine oszillierende Größenänderung hervorgehoben (siehe Abb. 6-12). Ist der betreffende Textblock im Vortrag animiert worden, kann der Zeitpunkt des Erscheinens des Textblockes in der Aufzeichnung dank der in 6.2.2 beschriebenen Nutzbarkeit animierter Folienelemente zur Navigation direkt angesprungen werden.

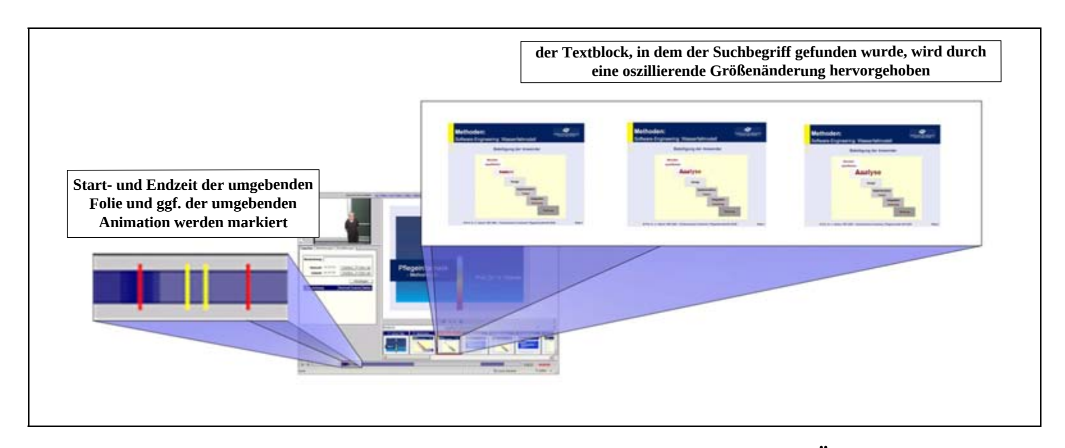



**Abb. 6-12: Verbindung der Volltextsuche mit zeit- und strukturbasierten Überblickskomponenten**

In der Zeitleiste werden jeweils die Grenzen der Folie, auf der das Suchresultat gefunden wurde, rot markiert. Wie in Abb. 6-12 dargestellt, werden bei im Vortrag animierten Suchresultaten darüber hinaus die Grenzen der Animation durch gelbe Markierungen kenntlich gemacht.
Die Volltextsuche ist auf diese Art mit beiden Überblickselementen verbunden und kann zusammen mit den damit verbundenen Navigationshilfen genutzt werden.

[68] In [Day, Foley, Groeneweg & Van der Mast 2005] wird eine Möglichkeit, Fragen an einzelne Folien koppeln zu können als Anforderung an Technologien zu Wiedergabe aufgezeichneter Vorlesungen herausgearbeitet. Das zum Zweck der dort vorgestellten Studie eingesetzte System verfügt nicht allerdings über eine solche Möglichkeit.

### 6.2.5 Backtracking

Backtracking wird, wie in 4.5.1.4 hergeleitet, als zeitbasierte Navigationshilfe realisiert. Bei jeder Navigationsaktion wie dem Anklicken eines strukturellen Elementes, dem Aufruf eines Bookmarks oder der Anwahl eines Punktes auf der Zeitleiste wird die Abspielposition des Videos vor dem Ausführen der Navigationsaktion in einem Stack gespeichert (vgl. 4.4.2.4 zur Funktionsweise von Backtracking Stacks in klassischen Hypermedien). Um Nutzern bei Betätigen des Backtracking-Knopfes ein Wiedererkennen der jeweiligen durch das Backtracking zurückgenommenen Navigationsaktion zu ermöglichen, wird ein Zeitversatz von drei Sekunden von der angesprungenen Position abgezogen. Bei mehrmaligem Betätigen des Backtracking-Knopfes hintereinander wird der jeweils oberste Eintrag des Backtracking-Stacks auf einen Forward-Stack gelegt, um wie bei einem herkömmlichen Webbrowser auch Backtracking-Aktionen zurücknehmen zu können.

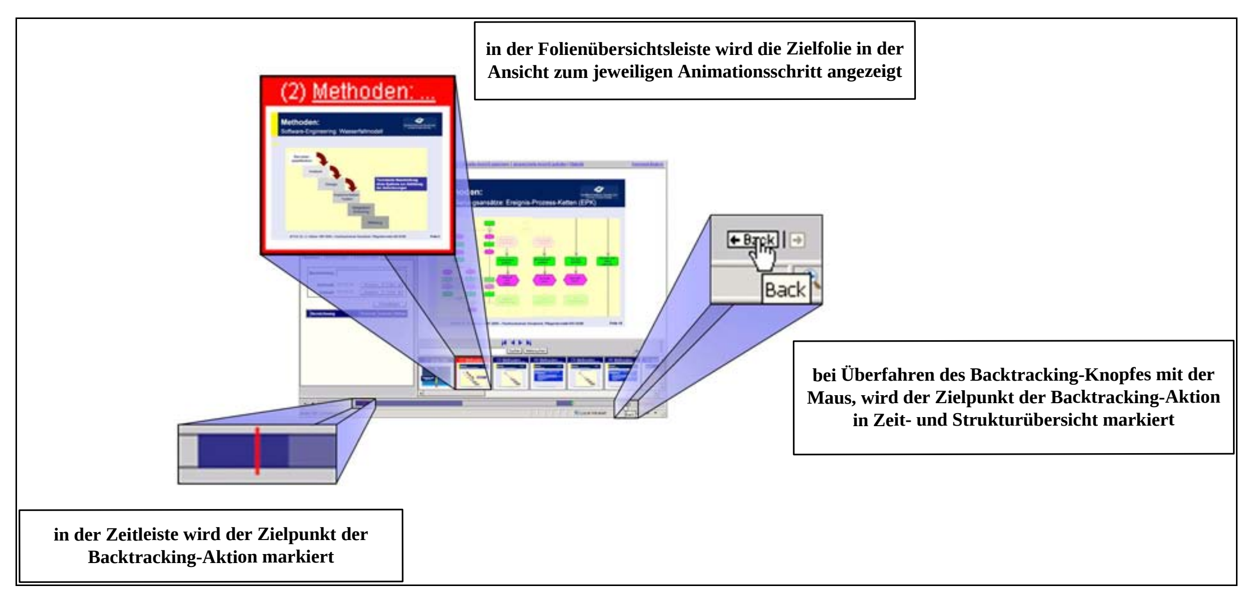



**Abb. 6-13: Verbindung von Backtracking mit zeit- und strukturbasierten Übersichtselementen**

Die Verbindung des Backtracking mit der strukturellen Übersicht wird, wie in Abb. 6-13 gezeigt, durch eine rote Umrandung der jeweils anzunavigierenden Folie in der Übersicht bei Überfahren des Backtracking-Knopfes mit dem Mauszeiger gegeben. Weicht die Darstellung der Folie zu dem betreffenden Zeitpunkt von der standardmäßig in der Folienübersicht angezeigten Endansicht ab, wird die Folienansicht so aktualisiert, dass in der Übersicht die Folienansicht zum anzusteuernden Zeitpunkt zu sehen ist (vgl. Abb. 6-4 auf Seite 114). In der Zeitleiste wird der durch die Backtracking-Aktion angesteuerte Zeitpunkt durch eine senkrechte rote Linie markiert.

## 6.3 Weiterführende Ansätze

Neben den bereits beschriebenen Standard-Navigationshilfen für Hypermedien existieren eine Reihe weiterführender Ansätze. Diese Ansätze sind dabei nicht als Alternativen zu den voran beschriebenen Navigationshilfen konzipiert, sondern sind vielmehr als eine Navigationsebene zu verstehen, die auf den in 6.2 beschriebenen Ansätzen aufbaut. Da im Rahmen der vorliegenden Arbeit bereits ein komplettes hypermediales Navigationskonzept für Vorlesungsaufzeichnungen entwickelt wird, soll im Folgenden kurz die Eignung verschiedener weiterführender hypermedialer Navigationsansätze für den Bereich Vorlesungsaufzeichnungen untersucht werden. Aufzeichnungen

### 6.3.1 Guided Tours

Guided Tours bieten Nutzern vordefinierte Präsentationsreihenfolgen von hypermedialen Knoten und erlauben so beispielsweise die Nutzung ein und derselben Materialsammlung durch Nutzer mit unterschiedlichen Vorkenntnissen. Im Kontext universitärer Lehre ist diese Form der Navigationsunterstützung allerdings selbst für klassische Hypermedien kaum verbreitet. Eine Reihe möglicher Gründe hierfür werden in [Brusilovsky & Miller 2001] diskutiert. Im Kontext von Vorlesungsaufzeichnungen spricht neben den von Miller und Brusilovsky diskutierten allgemeinen Erwägungen vor allem der Umstand, dass eine Vorlesung ohnehin schon eine Art vorstrukturierter Wissenspräsentation ist, gegen die Umsetzung des Konzepts von guided Tours für das Medium. Hinzu kommt die Tatsache, dass die Erstellung von guided Tours in hohem Maße mit zusätzlichem manuellem Arbeitsaufwand verbunden ist.

### 6.3.2 Adaptive Hypermedien und soziale Navigation

Einen weitaus elaborierteren Ansatz zur Navigationsunterstützung in Hypermedien stellen adaptive Hypermedien dar. Adaptive Hypermedien verknüpfen Ergebnisse aus den Forschungsbereichen Benutzermodellierung und Hypermedien, um die Präsentation von hypermedialen Inhalten gezielt auf den jeweiligen Nutzer zuschneiden zu können [Brusilovsky 2001]. So können beispielsweise bei der Darstellung von Webseiten einzelne Unterknoten hinzugefügt oder ausgelassen werden. Auch können Links auf andere Seiten je nach Kenntnisstand und Interessen des Nutzers angezeigt oder entfernt werden [Brusilovsky 2004]. Eine spezielle Form von adaptiver Navigationsunterstützung in Hypermedien ist soziale Navigation [Brusilovsky 2004]. Das Konzept sozialer Navigation in Hypermedien und in elektronischen Medien allgemein basiert auf dem Grundgedanken, dass die Nutzung des Mediums wie in der realen Welt auch, Spuren auf dem jeweiligen Medium hinterlässt [Wexelblat 1998]. Diese Spuren können später anderen Nutzern, analog zu Gebrauchsspuren beispielsweise bei Büchern in einer Bibliothek, Hinweise darauf liefern, welche Teile des Mediums von anderen Nutzern rezipiert wurden und wie intensiv dabei mit den jeweiligen Teilen gearbeitet wurde [Dieberger, Dourish, Höök, Resnick, Wexelblat 2000]. Zur sozialen Navigation in elektronischen Medien können darüber hinaus weitere Informationen, wie die Zugehörigkeit von Nutzern zu bestimmten Gruppen oder auch die zeitliche Einordnung von Medienzugriffen verwendet werden. Im eLearning sind Anwendungen, die auf sozialer Navigation beruhen vergleichsweise neu. Beispiele wie Knowledge Sea II zeigen jedoch, dass soziale Navigation dort mit Erfolg eingesetzt werden kann [Brusilovsky, Chavan & Farzan 2004].

Soziale Navigation ist im Gegensatz zu guided Tours und vielen Ansätzen im Bereich adaptive Hypermedien nicht durch einen Mehraufwand seitens des Contentproduzenten (in eLearning-Szenarien der Dozent oder seine Mitarbeiter) gekennzeichnet, da die Aufbereitung der Materialen durch die Interaktion der Nutzer (die Studierenden) mit den angebotenen Inhalten geschieht.

Zur Erfassung von Medienzugriffen werden bei Ansätzen zur sozialen Navigation in klassischen Hypermedien in der Regel Navigationsaktionen der Nutzer protokolliert. In [Farzan & Brusilovsky 2005] wird ein Ansatz vorgestellt, in dem die Zeiten zwischen zwei Navigationsaktionen als zusätzlicher Indikator für die Intensität der Nutzung der jeweils besuchten Knoten verwendet werden. So ist es nicht nur möglich, besonders intensiv genutzte Knoten zu markieren, sondern auch irrtümlich besuchte Knoten aufgrund extrem kurzer Verweilzeiten zu identifizieren und auszuschließen. Da im virtPresenter mit der in 6.2.1 beschriebenen Umsetzung des Footprint Konzeptes bereits eine Komponente zur zeitbasierten Erfassung von Nutzerinteraktionen implementiert ist, kann eine Speicherung dieser Daten auch zum Zweck sozialer Navigation vergleichsweise einfach realisiert werden. Durch die

Anbindung einer Komponente zur sozialen Navigation an das Footprint-Element und damit die Zeitleiste ist darüber hinaus eine Integration einer sozialen Navigationskomponente in das Gesamtnavigationskonzept realisierbar.

### 6.3.3 Fazit

Bei der vorangehenden Betrachtung erweist sich vor allem soziale Navigation als geeigneter und leicht in das vorhandene Navigationskonzept integrierbarer Ansatz. Soziale Navigation bedarf keines zusätzlichen Wartungs- und Erstellungsaufwandes und kann sowohl technisch als auch konzeptuell in das in 6.2 beschriebene Navigationskonzept integriert werden.
Dieser Überlegung folgend wurde im Rahmen der vorliegenden Arbeit ein Konzept zur Integration sozialer Navigation entwickelt, das im nächsten Abschnitt dargestellt wird.

## 6.4 Ein Konzept zur Integration sozialer Navigation in Vorlesungsaufzeichnungen

Wie bereits im vorangehenden Abschnitt erarbeitet, ist es sinnvoll ein Konzept zur sozialen Navigation in Vorlesungsaufzeichnungen auf der Footprint Funktionalität des in 6.2 dargestellten hypermedialen Navigationskonzeptes für Vorlesungsaufzeichnungen aufzubauen. Dieser Abschnitt stellt eine derart zeitbasierte Adaption sozialer Navigation vor und diskutiert deren Einpassung in das Gesamtinterface. Teile der hier entwickelten Überlegungen sowie das vorgestellte Konzept wurden bereits in [Mertens, Farzan & Brusilovsky 2006] veröffentlicht.

### 6.4.1 Integration

Soziale Navigation ist als eine modulare Komponente realisiert, die direkt über der Zeitleiste eingeblendet werden kann (siehe Abb. Abb. 6-14).

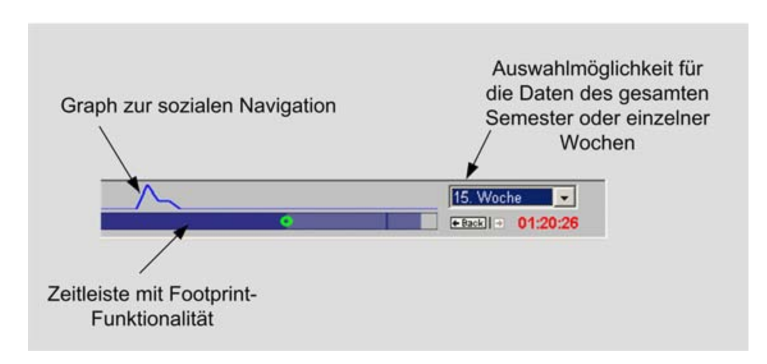



**Abb. 6-14: Anordnung der Komponente zur sozialen Navigation über der Zeitleiste**

In Form eines Graphen werden die Interaktionsdaten aller Benutzer mit Ausnahme des aktuellen Benutzers in der selben zeitlichen Auflösung wie die Zeitleiste visualisiert. Bereiche, die durch persönliche Footprints des Nutzers markiert sind, können damit direkt mit den Interaktionsspuren anderer Nutzer verglichen werden. Auch können so die in 6.1 und 6.2 beschriebenen Mechanismen zu Verbindung von zeit- und strukturbasierter Navigation zusammen mit der sozialen Navigationskomponente genutzt werden. Direkt neben dem Graphen ist eine Auswahlbox angeordnet, die es ermöglicht, den Graphen für einzelne Semesterwochen oder die kumulierten Daten anzuzeigen (vgl. 6.4.4).

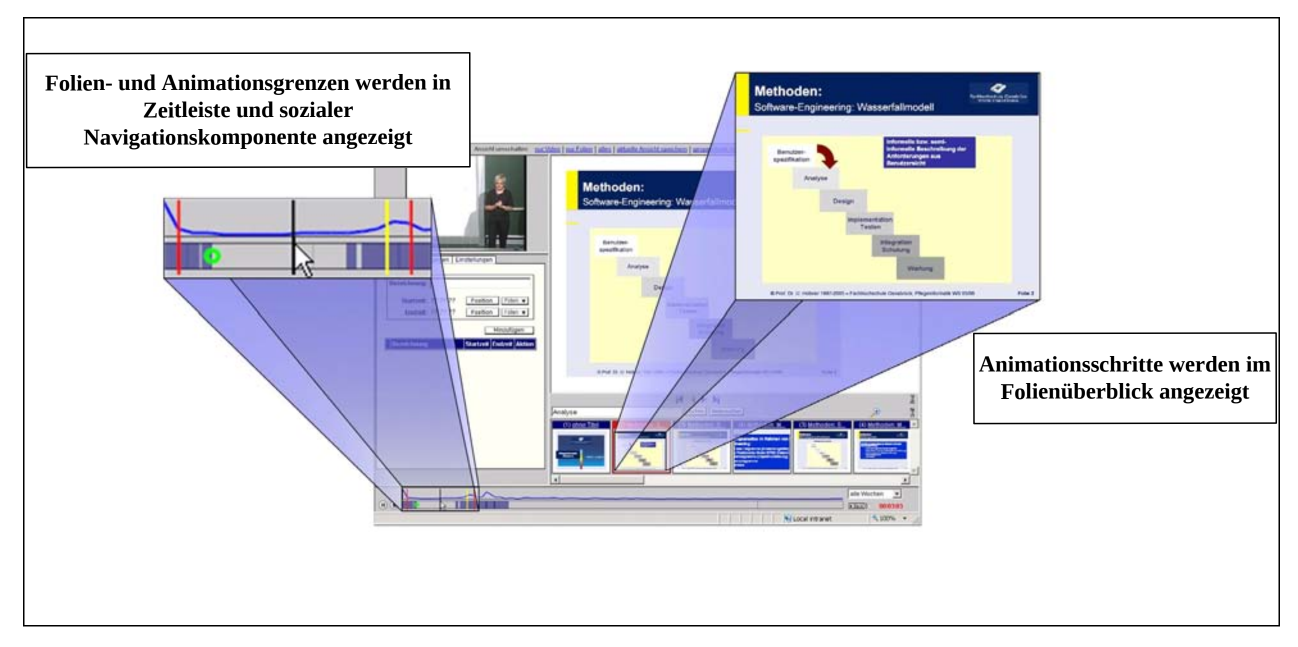



**Abb. 6-15: Markierungen in der Zeitleiste werden auch in der Komponente zur sozialen Navigation angezeigt.**

Wie in Abb. 6-15 dargestellt, werden bei Navigation in der Zeitleiste, die Markierungen von Folien- und Animationsgrenzen auf den Graphen ausgeweitet. Diese Ausweitung der Markierungen ist auch bei allen anderen Navigationshilfen realisiert, die mit der Zeitleiste verbunden sind. So können beliebige Bereiche des Graphen auch mit dem strukturellen Überblick der Vorlesung oder auch Bookmarks, Volltextsuche usw. in Verbindung gebracht werden.
Bei zeitbasierter Navigation in der Zeitleiste wird auch die Markierung der jeweils angesteuerten Position in der Zeitleiste und in der sozialen Navigationskomponente angezeigt. So können im Graphen markierte Regionen direkt über die Zeitleiste annavigiert werden.

## 6.4.2 Visualisierung

Die Visualisierung der Interaktion anderer Nutzer mit der Aufzeichnung als Graphen wurde gewählt, um eine optische Abgrenzung von der Footprint-Komponente des Interfaces zu erreichen. Damit ist die Wahrscheinlichkeit einer Verwechslung beider Komponenten geringer. Weitere Gründe, die an dieser Stelle für die Verwendung eines Graphen sprechen sind die Möglichkeit einer höheren Auflösung der Darstellung und eine einfachere Skalierbarkeit. Im Gegensatz zur Markierung des Nutzungsgrades durch zunehmende Farbintensität sind Höhenunterschiede leichter wahrnehmbar. Zur Skalierung des Graphen wird der höchste Wert der Interaktionsfunktion (siehe 6.4.3) jeweils im Maximum des Darstellungsbereiches dargestellt. Alle anderen Werte werden linear skaliert.
Die durch den Graphen ermöglichte bessere Differenzierung zwischen Nutzungsintensitäten im Vergleich zur Footprint-Funktionalität für Einzelnutzer ist aufgrund der unterschiedlichen Eingangsdaten nötig. In der Footprint-Leiste ist vor allem die Aussage, ob eine Passage bereits angeschaut wurde entscheidend. Zur Prüfungsvorbereitung ist es darüber hinaus relevant, ob eine Passage nur einmal oder mehrmals angesehen wurde. Ob sie drei-, vier- oder fünfmal wiederholt wurde, ist in der Regel irrelevant. Zur sozialen Navigation ist hingegen

ein Eindruck der absoluten Zahlen der Zugriffe im Verhältnis wichtig. So können Passagen mit insgesamt zwei Zugriffen schlechter von Passagen mit einem Zugriff unterschieden werden, der Unterschied zwischen keinem, dreißig und hundert Zugriffen ist aber klarerer erkennbar. Daher bietet sich in diesem Zusammenhang eine lineare Skalierung der Daten zur Visualisierung an.

### 6.4.3 Berechnung der Interaktionsfunktion

Wie bereits in 6.4.1 erwähnt, werden in der sozialen Navigationskomponente nur die Zugriffe anderer Nutzer visualisiert. Dadurch kann der Nutzer erkennen, wenn Bereiche der Aufzeichnung nur von ihm selbst betrachtet wurden, nicht aber von anderen Nutzern. In diesem Fall ist der Bereich zwar in der Footprint-Leiste markiert, nicht jedoch in der sozialen Navigationskomponente. Diese Information kann beispielsweise bei der Kommunikation über Kursinhalte von Bedeutung sein. Verfügt ein Nutzer über Wissen, das den anderen Kursteilnehmern nicht bekannt ist, sind derartig gekennzeichnete Bereiche potentiell die Quelle dieses Wissens.
Die Unterscheidung von Footprint-Leiste und sozialer Navigationskomponente kann umgekehrt auch genutzt werden, um Wissenslücken des Nutzers zu lokalisieren. Sind Bereiche in der sozialen Navigationskomponente hervorgehoben und in der Footprint-Leiste als unbekannt markiert, kann dort Information vorhanden sein, die von anderen Kursteilnehmern als wichtig erachtet wurde.
Zur Identifikation von Bereichen der Aufzeichnung, die von anderen Nutzern betrachtet wurde und für den augenblicklichen Nutzer neu sind, reicht ein einfacher visueller Vergleich von Footprint-Leiste und sozialer Navigationskomponente. Zur Identifikation von Bereichen die nur der augenblickliche Nutzer betrachtet hat, ist jedoch wie eingangs erwähnt der Ausschluss des Nutzungsverhaltens des aktuellen Nutzers aus der Berechnungsfunktion der Interaktion anderer Nutzer mit der Aufzeichnung sinnvoll. Würden die Daten des aktuellen Nutzers in die Darstellung mit einbezogen, wären sowohl Footprints als auch soziale Navigationskomponente an der betreffenden Stelle markiert. Die Berechnung der Interaktionsfunktion ist in Formel 6-1 beschrieben.

$$f_j(t) = \sum_{i=1}^{m} u_i(t) - u_j(t)$$

**Formel 6-1: Berechnungsfunktion der Anzeige zur sozialen Navigation für Nutzer *j* zum Zeitpunkt *t***

Mit der Formel wird die Interaktionsfunktion für den Nutzer $j$ zum Zeitpunkt $t$ der Aufzeichnung berechnet. Dabei steht $j$ für den aktuellen Nutzer und $m$ für die Anzahl aller Nutzer. Die Funktion $u$ gibt die Interaktion eines Nutzers für den Zeitpunkt $t$ an. Hat ein Nutzer die Aufzeichnung zum Zeit Punkt $t$ nicht betrachtet, ist $u(t)$ null, hat er sie $n$ mal betrachtet, ist $u(t)=n$.

### 6.4.4 Filterung der Interaktionsdaten

Im Laufe eines Semesters werden Vorlesungsaufzeichnungen zu unterschiedlichen Zwecken eingesetzt [Zupancic & Horz 2002]. So wird die Aufzeichnung während des Semesters je nach Einsatzszenario beispielsweise zur Nachbereitung der Veranstaltung oder als Wissensquelle bei der Durchführung von Übungsaufgaben genutzt. In der vorlesungsfreien Zeit dienen sie zur Vorbereitung von Prüfungen und Klausuren.
Um diese Tatsache widerzuspiegeln, kann die Anzeige der Interaktionsdaten anderer Nutzer auf einzelne Wochen des Semesters beschränkt werden. Im Gegensatz zur Standardanzeige werden dann nur die Interaktionsdaten angezeigt, die in der jeweils ausgewählten Woche

gesammelt wurden. Die Interaktionsdaten werden bei Nutzung der Aufzeichnung über mehrere Semester akkumuliert. Semesteranfangstermine und ggf. Abweichungen für einzelne Veranstaltungen werden zu diesem Zweck in einer Datenbank verwaltet.
Weitere Möglichkeiten zur Filterung sind denkbar. So könnten beispielsweise Ähnlichkeitsmaße zum Vergleich von Interaktionsdaten verschiedener Nutzer zur Bildung von Nutzergruppen mit ähnlichem Kenntnisstand verwendet werden. Auch könnten statistische Verfahren herangezogen werden, um untypisches Nutzerverhalten, das beispielsweise durch Netzwerkprobleme entstehen kann aus der Berechnung auszuschließen. In der aktuellen Version ist jedoch nur die oben beschriebene Filterung nach dem Kriterium der Semesterwoche implementiert.

### 6.4.5 Fazit

Die in diesem Abschnitt vorgestellte Erweiterung des in 6.2 beschriebenen hypermedialen Navigationskonzeptes zeigt, dass soziale Navigation in Vorlesungsaufzeichnungen aufbauend auf dem im Rahmen dieser Arbeit entwickelten Navigationskonzept realisiert werden kann. Aufgrund der zeitbasierten Konzeption der vorgestellten sozialen Navigationskomponente kann diese nahtlos in das Gesamtinterface integriert werden.
Soziale Navigation kann dabei in Vorlesungsaufzeichnungen fast genau so verwendet werden wie in klassischen Hypermedien mit dem Unterschied dass Interaktionsdaten zeitbasiert erfasst und dargestellt werden. Ein Vorteil der zeitbasierten Erfassung von Interaktionsdaten allgemein wird bei eingehenderer Betrachtung des in [Farzan & Brusilovsky 2005] vorgestellten Ansatz zu sozialen Navigation in klassischen Hypermedien deutlich. Dort wird die Verweilzeit eines Nutzers auf einer Webseite als Indikator für den Nutzungsgrad der jeweiligen Interaktion verwendet. Bei Medien mit einem video- oder audiobasierten Anteil liegt die Vermutung nahe, dass die Validität eines zeitbasierten Interaktionsmaßes sogar höher liegt, da Nutzer die Aufzeichnung eher stoppen, wenn sie zu einer anderen Aufgabe wechseln, um eine Ablenkung durch die Tonspur zu vermeiden.
Insgesamt lässt sich festhalten, dass das in diesem Abschnitt vorgestellte Konzept zur sozialen Navigation in Vorlesungsaufzeichnungen die Kernfunktionalität einer sozialen Navigationskomponente deutlich demonstriert. An einigen Stellen weist die Implementation jedoch auch Möglichkeiten zur Weiterentwicklung auf. Insbesondere bei der Aufbereitung und Filterung gesammelter Interaktionsdaten sind hier weiterführende Forschungsperspektiven gegeben.

## 6.5 Fazit und Einordnung in den weiteren Verlauf

Das Kapitel hat eine Möglichkeit zur Umsetzung der in Kapitel 4 erarbeiteten Anforderungen an ein hypermediales Navigationskonzept für Vorlesungsaufzeichnungen aufgezeigt und eine darauf basierende Implementation vorgestellt. In Abschnitt 6.4 wurde gezeigt, dass auf dieser Implementation aufbauend weiterführende Navigationshilfen wie soziale Navigation realisiert werden können. Auch hierzu wurde eine Implementation vorgestellt.
Im folgenden Kapitel wird die Implementation des Aufzeichnungssystems und des Wiedergabeinterfaces aus technischer Sicht beschrieben. Dabei wird unter anderem diskutiert, wie die hier vorgestellten Funktionalitäten technisch umgesetzt sind.

# 7 Implementation des Systems

Wie bereits in Kapitel 2 dargestellt, lässt sich die Erstellung und Nutzung von Vorlesungsaufzeichnungen in vier Phasen gliedern. Im Einzelnen sind dies Vorbereitung, Aufzeichnung, Nachbearbeitung und Nutzung. Das im Rahmen dieser Arbeit entwickelte System zur Vorlesungsaufzeichnung und Wiedergabe besteht aus mehreren Komponenten, die sich den einzelnen Phasen zuordnen lassen.

Dieses Kapitel ist daher anhand des in 2.1 beschriebenen Vier-Phasen-Modells aufgebaut. In 7.1 wird eine Komponente zur automatisierten Vorbereitung der Aufzeichnungsmaterialien vorgestellt. In 7.2 werden die zur Aufzeichnung selbst verwendeten Softwarekomponenten beschrieben. Daran anschließend wird in 7.3 die Prozesskette zur Nachbearbeitung der aufgezeichneten Daten dargestellt. In 7.4 wird die Implementation des Wiedergabe-Interfaces behandelt. Einzelheiten zur Erzeugung und Aufbereitung der zur Wiedergabe verwendeten SVG-Folien während der Nachbereitungsphase werden aufgrund der Komplexität der durchzuführenden Arbeitsschritte gesondert in Kapitel 8 diskutiert.

In den Abschnitten 7.1 bis 7.4 wird jeweils zunächst einleitend die Grundproblematik der jeweiligen Phase und der entwickelte Lösungsansatz behandelt. Daran anschließend werden jeweils im ersten Unterabschnitt Grundlagen wie beispielsweise die Auswahl der verwendeten Programmiersprachen und ggf. Datenformate diskutiert. In dem darauf folgenden Unterabschnitt werden relevante Details der jeweiligen Implementation dargestellt.

Dieser Abschnitt stellt dabei keine Einführung in die jeweils verwendeten Programmiersprachen und Dateiformate dar. Es werden lediglich implementationsrelevante Punkte behandelt. Für umfassende Darstellungen zum Umgang mit den verwendeten Programmiersprachen und Formaten sei an dieser Stelle auf die entsprechende Literatur wie beispielsweise [Robinson, Nagel, Glynn, Skinner, Watson & Evjen 2004] oder [Albahari, Drayton & Merrill 2001] zu C#, [Hall 2002] zu JSP und [Watt & Lilley 2002] oder [Eisenberg 2002] zu SVG verwiesen.

Eine verkürzte Darstellung von Teilbereichen der hier beschriebenen Implementation ist [Mertens, Ketterl & Vornberger 2007] veröffentlicht.

## 7.1 Vorbereitungsphase

Zur Vorbereitung einer Aufzeichnung werden in vielen PowerPoint-basierten Systemen die verwendeten Folien in ein von der Aufzeichnungssoftware verwendetes, proprietäres Format konvertiert. Bei Verwendung derartiger Zwischenformate kann es allerdings leicht zu Darstellungsfehlern oder zu Beeinträchtigungen bei der Präsentation wie beispielsweise fehlerhafter Wiedergabe von Animationen kommen (vgl. 5.2). Das im Rahmen dieser Arbeit entwickelte System verwendet daher ähnlich wie der in [Mühlhäuser & Trompler 2002] vorgestellte Ansatz ein Listener-Konstrukt, das es ermöglicht, PowerPoint zur Präsentation der verwendeten Folien zu nutzen. Ereignisse wie Folienwechsel oder auch Animationsschritte werden dabei von einer externen Applikation protokolliert.

Die Voraussetzung zur Verwendung eines derartigen Listener-Konstruktes ist von Microsoft durch die Bereitstellung einer COM-Schnittstelle gegeben, über die verschiedene PowerPoint-Ereignisse verarbeitet werden können [20]. Diese Schnittstelle ist allerdings mit dem Nachteil behaftet, dass bei der Wiedergabe von Animationsschritten nicht eindeutig festgestellt werden kann, um welchen es sich jeweils handelt. Zwar verfügen Animationsschritte in einer PowerPoint-Präsentation über eindeutige ID-Werte, bei Ablauf einer Bildschirmpräsentation werden jedoch von PowerPoint zur Laufzeit erzeugte Kopien der jeweiligen Animationsschritte verwendet, die wieder neue IDs zugewiesen bekommen.

Um dieses Problem zu umgehen, wird an jeden Animationseffekt in der Präsentation ein weiterer gekoppelt. Dieser zusätzlich in die Präsentation eingefügte Animationseffekt

verfügt über ein minimal kleines Textfelds. Jedes dieser Textfelder wird mit einer eindeutigen ID als Inhalt versehen. Aufgrund ihrer geringen Größe sind die Textfelder in der Präsentation nicht sichtbar. Anschließend werden beide Animationseffekte so miteinander gekoppelt, dass sie gleichzeitig ausgeführt werden. Der zusätzlich eingefügte Effekt wird damit eindeutig dem zu identifizierenden Folienobjekt zugeordnet und dient so zur Identifikation des jeweiligen Animationseffektes. Die Implementation dieser Schritte ist bei Verwendung geeigneter Schnittstellen zu PowerPoint vergleichsweise einfach realisierbar. Im Folgenden werden daher in 7.1.1 zunächst Grundlagen der Fernsteuerung von PowerPoint erläutert. In 7.1.2 werden kurz Details der Implementation dargestellt.

### 7.1.1 Grundlagen

PowerPoint kann von verschiedenen Programmiersprachen wie beispielsweise C, C++, C# oder Visual Basic aus ferngesteuert werden [23]. Eine Interaktion mit PowerPoint kann dabei direkt über die COM-Schnittstelle von PowerPoint oder aber aus dem .NET Framework heraus realisiert werden.
In den meisten Fällen bietet sich die Verwendung des .NET Frameworks an, da die dort verwendeten Programmiersprachen wie C# eine Reihe von Vorteilen wie beispielsweise Objektorientierung, Typsicherheit, automatische Speicherverwaltung und sicheres Ausführungsverhalten mit sich bringen [23].
Im Rahmen dieser Arbeit wird C# zur Implementation der Interaktion mit PowerPoint und einer Reihe weiterer Aufgaben verwendet. Gründe für diese Entscheidung werden in 7.1.1.2 diskutiert. In 7.1.1.1 werden die Grundlagen der Interaktion zwischen .NET und PowerPoint dargestellt.

#### 7.1.1.1 Interaktion zwischen .NET und PowerPoint

Die Interaktion zwischen im .NET Framework programmierten Anwendungen und PowerPoint findet nicht direkt über die COM-Schnittstelle von PowerPoint sondern über ein zusätzliches Wrapper-Konstrukt statt. Durch die Verwendung des Wrappers bleibt die Nutzung von COM zwar in der Programmierung transparent, es können jedoch trotzdem Probleme auftreten, die durch die Verwendung von COM bedingt sind. In 7.1.1.1.1 wird daher zunächst ein grober Einblick in die Arbeitsweise des der Kommunikation mit PowerPoint zugrunde liegenden Component Object Models (COM) gegeben.

##### 7.1.1.1.1 Prozesskommunikation über das Component Object Model

Microsofts Component Object Model (COM) ermöglicht die Kommunikation mit Anwendungen von externem Programmcode aus. Damit kann die Anwendung ferngesteuert und Objekte innerhalb der Anwendung erstellt und manipuliert werden [Rubin & Bain 1999]. Ähnlich wie bei einem verteilten System werden diese Objekte im jeweiligen Programm durch Stubs vertreten. Die Prozesskommunikation verläuft dabei für den Programmierer transparent. In Analogie zu verteilten Systemen, wird ein Programm, das über die COM-Schnittstelle einer Applikation kommuniziert, COM-Client und die Applikation COM-Server genannt. Im Kontext der Fernsteuerung von Office-Applikationen wie beispielsweise PowerPoint oder Word werden die COM-Server dieser Applikationen auch Automation-Server genannt.

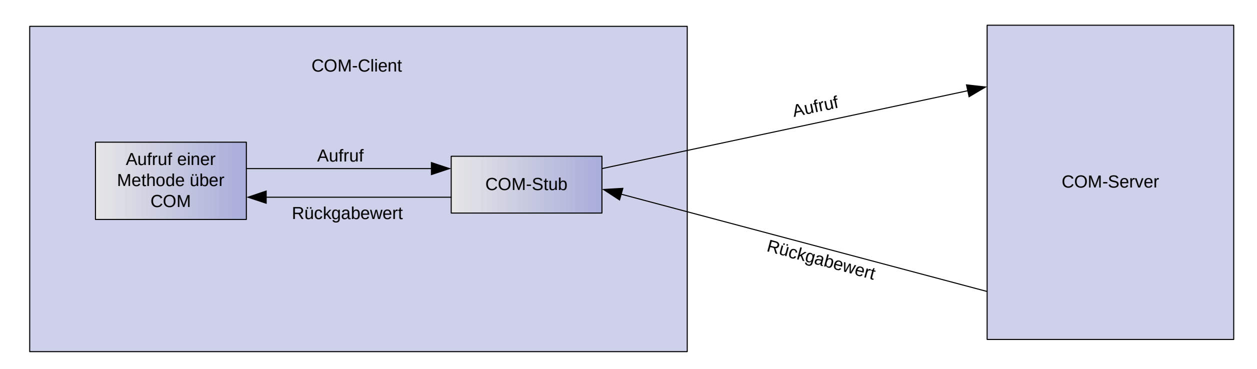



**Abb. 7-1: Methodenaufrufe über COM**

In Abb. 7-1 wird deutlich, wie ein Methodenaufruf über COM im Fall der Kommunikation mit PowerPoint funktioniert. Der Stub serialisiert eventuelle Aufrufparameter und leitet den Aufruf über die Prozessgrenze hinweg zum Server weiter. Im COM-Server wird der Methodenaufruf verarbeitet, eventuelle Rückgabewerte werden ebenfalls serialisiert und an den Stub übermittelt. In der Terminologie verteilter Systeme werden Umwandlung und Transport von Daten in diesem Kontext Marshalling genannt [Robinson, Nagel, Glynn, Skinner, Watson & Evjen 2004].

#### 7.1.1.1.2 Runtime Callable Wrappers und das .NET Framework

Auch im Rahmen des .NET Frameworks ist es möglich, Applikationen über deren COM-Schnittstelle fernzusteuern. Im Code ist dabei zwischen Aufrufen über COM und Methodenaufrufen nativer .NET Typen kein Unterschied zu erkennen [Troelsen 2002]. Diese Transparenz wird durch den Einsatz so genannter Runtime Callable Wrappers (RCW) erreicht [Troelsen 2002]. Wie in Abb. 7-2 verdeutlicht, stellen RCWs einen weiteren Zwischenschritt dar. Dieser Zwischenschritt ist ein notwendiges Bindeglied, da .NET im Gegensatz zu den Sprachen, die COM unterstützen, mit verwaltetem Speicherbereich arbeitet. Auch beim Übergang von verwaltetem zu nicht verwaltetem Speicherbereich findet wieder Marshalling statt, das jedoch nicht mit dem in 7.1.1.1.1 erwähnten Marshalling bei der Kommunikation über Prozessgrenzen hinweg gleichzusetzen ist [Gordon 2002].

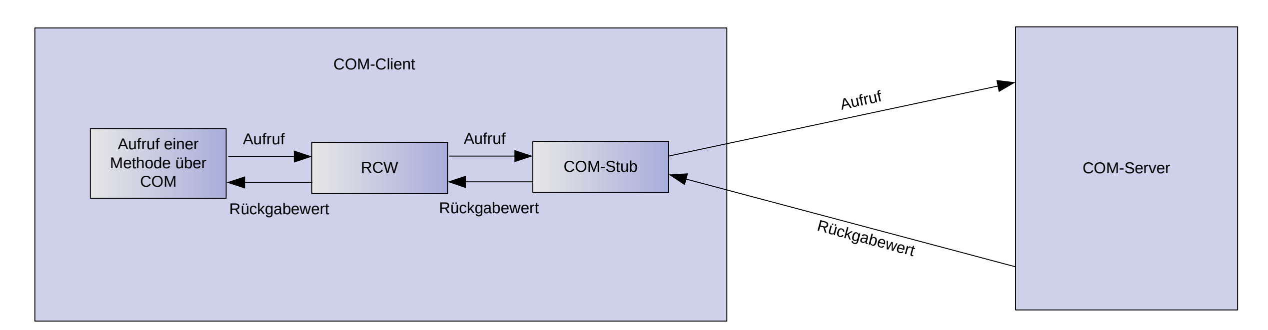



**Abb. 7-2: Runtime Callable Wrappers als Bindeglied zu COM**

Eine spezielle Art von RCW's sind die so genannten Primary Interop Assemblies. Primary Interop Assemblies werden vom Ersteller einer COM Komponente mit der Komponente zusammen ausgeliefert und sind speziell auf die Komponente abgestimmt [Robinson, Nagel, Glynn, Skinner, Watson & Evjen 2004].

Auch für die Applikationen der Microsoft Office-XP Familie existieren derartige Primary Interop Assemblies, die auf einer Webseite zum Download bereitstehen [39]. Über die Primary Interop Assemblies kann der Automation-Server der jeweiligen Applikation direkt angesprochen werden. Der Automation-Server einer Applikation der Office Familie ermöglicht wiederum die Steuerung der jeweiligen Applikation durch ein Programm. Daher ist es möglich, beliebige Applikationen der Office-XP-Familie und damit insbesondere auch

PowerPoint aus einem externen .NET-Programm heraus zu steuern. Im Gegensatz zur Steuerung über ein internes Makro wird PowerPoint dabei als separater Prozess gestartet [12], was unter Umständen zu Problemen führen kann, wie in 7.3.2.2.2 beschrieben. Da es mit reinen Makros allerdings nicht realistisch möglich ist, Programme in der im Rahmen dieser Arbeit notwendigen Komplexität zu erstellen (vgl. 9.2.1), ist die Nutzung der COM-Schnittstelle von PowerPoint unumgänglich.

#### 7.1.1.2 C#

Das .NET Framework unterstützt eine Reihe verschiedener Programmiersprachen wie C#, C++ und Visual Basic. C# sticht aus den unterstützen Sprachen vor allem dadurch heraus, dass es eine sehr moderne Sprache ist, die speziell für die Verwendung im .NET Framework entwickelt wurde [Robinson, Nagel, Glynn, Skinner, Watson & Evjen 2004]. Vorteile gegenüber Visual Basic finden sich vor allem in der Strukturierbarkeit des zu entwickelnden Programmcodes. Im Vergleich zu C++ zeigt sich der Altersunterschied beider Sprachen. Dies wird beispielsweise an der Trennung der Deklaration und Definition von Objektfunktionen deutlich [Robinson, Nagel, Glynn, Skinner, Watson & Evjen 2004].
Unter der Perspektive der Wartbarkeit des entwickelten Codes bietet C# weitere Vorteile aufgrund seiner Ähnlichkeit zur weit verbreiteten Programmiersprache Java. Besonders im Umfeld der Universität Osnabrück finden sich eher Programmierer, die Java beherrschen als C++ oder Visual Basic. Im Rahmen der vorliegenden Arbeit wird daher C# zur Implementation der meisten Windows-basierten Komponenten verwendet.

### 7.1.2 Details der Implementation

Wie bereits eingangs erwähnt, werden bei der Vorbereitung einer PowerPoint-Datei zur Aufzeichnung alle Animationseffekte mit eindeutigen Markerobjekten in Form von Textfeldern versehen. Dabei können die eingefügten Textfelder nicht direkt mit dem jeweiligen Animationseffekt verknüpft werden, da in diesem Fall die Verbindung mit dem ursprünglich animierten Folienobjekt aufgehoben würde. Um dieses Problem zu vermeiden, werden die betreffenden Textfelder an einen Dummy-Animationseffekt gekoppelt, der zeitgleich mit dem jeweils zu verfolgenden Animationseffekt ausgeführt wird. Eine Verwechslung dieser zur Identifikation von Animationseffekten eingefügten Textfelder mit bereits ursprünglich in der Präsentation vorhandenen Textfeldern ist durch die Anordnung der Dummy-Animationseffekte auf der Zeitleiste ausgeschlossen. Dabei wird die Tatsache ausgenutzt, dass in PowerPoint alle Animationseffekte hintereinander auf einer Zeitleiste angeordnet sind.
Gleichzeitige Effekte sind durch ein spezielles Attribut gekennzeichnet. Ein Dummy-Animationseffekt wird immer direkt vor dem jeweils zu identifizierenden Animationseffekt positioniert. Dadurch, dass die in der Aufzeichnungskomponente verwendete Event-Schnittstelle von PowerPoint bei gleichzeitigen Effekten immer nur den in der Zeitleiste zu erst angeordneten Effekt übermittelt, werden so bei der Event-Verarbeitung nur die Folienelemente berücksichtig, die einem Dummy-Animationseffekt zugeordnet sind.
Die Zuordnung der in den Textfeldern gespeicherten ID-Werte zu den jeweiligen Original-Effekten wird durch Hashtabellen realisiert. Die Indizierung der Präsentation ist durch einen linearen Durchlauf implementiert, bei dem jeder Effekt einmal betrachtet wird. Abschließend ist festzustellen, dass die in der Vorbereitungsphase verwendete Software geringe Speicher- und Rechenkomplexität aufweist. Dies liegt vor allem in der Tatsache begründet, dass die Konvertierung der Präsentation in ein anderes Format in die Nachbearbeitungsphase verlagert wurde.

## 7.2 Live-Vortrag und Aufzeichnung

Wie in 2.1.2.2 beschrieben, ist die Aufzeichnung eines Live-Vortrags im Hörsaal in der Regel mit einer Reihe von zusätzlichen Aufwandsfaktoren verbunden. Hier ist in erster Linie der Auf- und Abbau der verwendeten Aufnahmetechnik zu nennen. Hinzu kommen damit verbundene Tätigkeiten wie Aufnahmetests und das Justieren von Kameras und Mikrofonen. Um diesen Zusatzaufwand zu vermeiden, ist die Aufzeichnung als Teil einer vollautomatisierten Prozesskette realisiert, die in die vorhandene Hörsaaltechnik integriert ist.
So findet die Aufzeichnung des Videomitschnittes nicht auf dem Dozentenrechner sondern auf einem Aufzeichnungsrechner statt. Dieser Aufzeichnungsrechner ist in der Regel im Technikraum des Hörsaals untergebracht und mit einer fest im Hörsaal verbauten Kamera verbunden. Dadurch, dass Kamera und Aufzeichnungsrechner permanent im Hörsaal verbleiben, sind Schritte wie Nachjustieren oder erneutes Abstimmen des Aufzeichnungsrechners auf die Kamera nach einmaliger Installation im Hörsaal nicht mehr nötig. Es muss lediglich eine TCP/IP-Verbindung zum Aufzeichnungsrechner hergestellt werden.

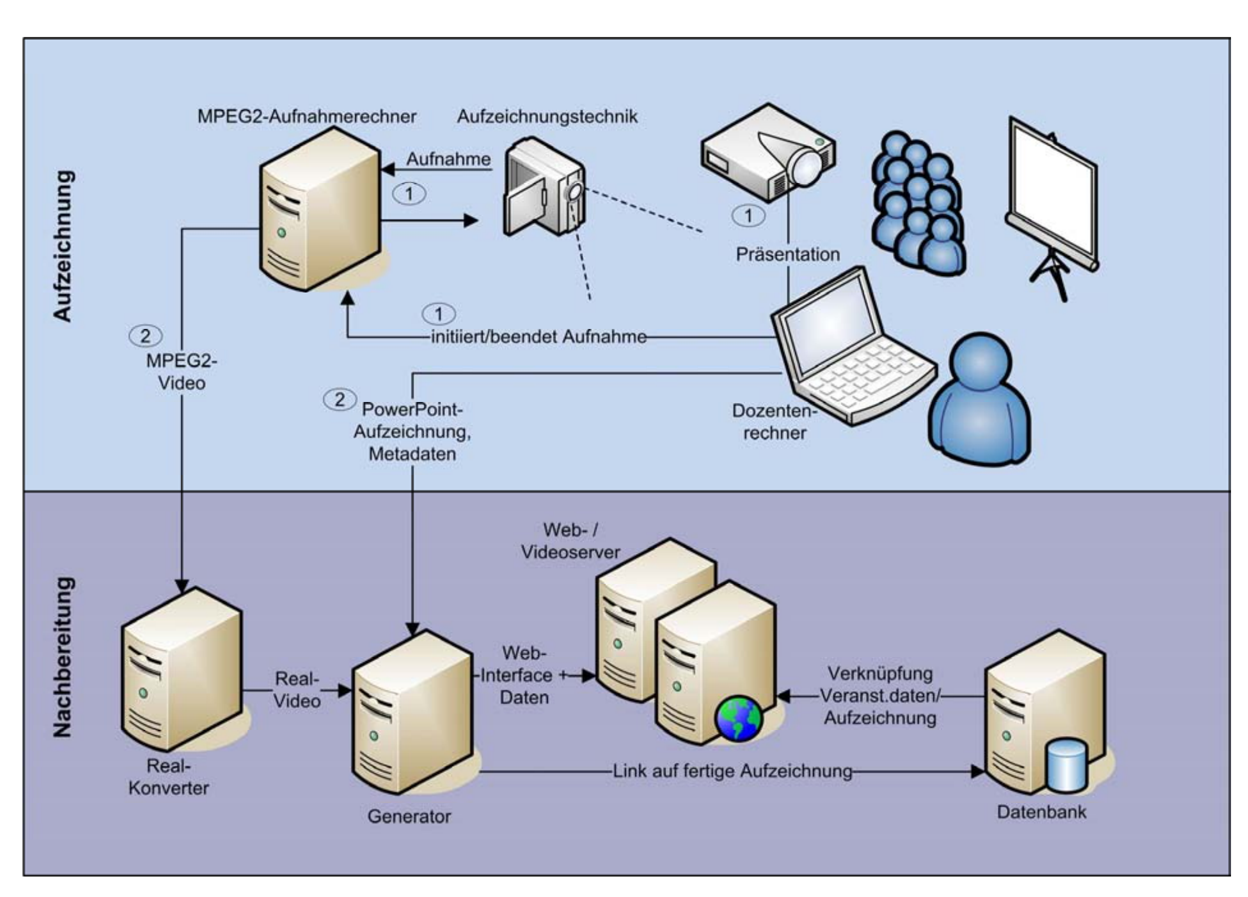



**Abb. 7-3: verteilte Vortragsaufzeichnung mit dem virtPresenter**

Zusätzlich zum Videomitschnitt werden die Zeitindices von Folienwechseln und Animationsschritten der Präsentation protokolliert, um diese im Wiedergabeinterface zeitsynchron darstellen zu können. Wie bereits in der Einleitung zu 7.1 erwähnt, wird zu diesem Zweck ein Listener-Konstrukt genutzt, das von PowerPoint über die COM-Schnittstelle der Applikation zur Verfügung gestellt wird.
Im Gegensatz zur Video-Aufzeichnung kann die Aufzeichnung der von PowerPoint generierten Events nur auf dem Präsentationsrechner des Dozenten stattfinden, da nur dort eine Möglichkeit besteht, auf die COM-Schnittstelle der Applikation zuzugreifen. Das dazu verwendete Aufzeichnungsprogramm arbeitet daher auf dem Rechner des Dozenten. Wie in Abb. 7-3 dargestellt, wird die Aufnahme auf dem Videoaufzeichnungsrechner durch die

Aufzeichnungsapplikation auf dem Dozentenrechner initiiert und später beendet (1). Da der Videoaufzeichnungsrechner im MPEG-2 Format aufzeichnet, ist er in der Abbildung als MPEG2-Aufzeichnungsrechner gekennzeichnet.
Nach Beendigung der Aufzeichnung, sendet der Dozentenrechner zunächst ein Signal an den MPEG2-Aufzeichnungsrechner, um die Videoaufnahme zu beenden. Anschließend sendet der Dozentenrechner die zur Aufzeichnung verwendete PowerPoint-Datei zur weiteren Verarbeitung an einen zentralen Server. Auch die aufgezeichnete Videodatei wird zur Konvertierung auf einen zentralen Server verschoben (2). Die darauf folgenden Schritte sowie der genaue Ablauf der Kommunikation zu den betreffenden Servern sind Teil der Nachbearbeitungsphase und werden daher in 7.3 eingehender behandelt.
Im aktuellen Abschnitt werden zunächst Grundlagen wie die Vorraussetzungen zu einer synchronisierten Aufzeichnung von Film und Folien sowie Details zum verwendeten Videoformat diskutiert. Daran anschließend wird in 7.2.2 die Implementation der zur Aufzeichnung verwendeten Softwarekomponente dargestellt.

## 7.2.1 Grundlagen

Da während der Wiedergabe Folien und Videoaufzeichnung synchron abgespielt werden sollen, müssen die Videoaufzeichnung und die Protokollierung der Folienaufzeichnung miteinander auf wenige Decisekunden genau synchronisiert werden. Aufgrund technischer Hindernisse ist eine derart genaue Synchronisation von Videoaufzeichnung und Protokoll einzelner Schritte der Präsentation jedoch nicht ohne weiteres möglich. In 7.2.1.1 werden verschiedene Ansätze zur Lösung dieses Problems diskutiert. In 7.2.1.2 werden Gründe für die Auswahl des MPEG-2 Formates sowie Hardwareanforderungen an den verwendeten Videoaufzeichnungsrechner erläutert.

### 7.2.1.1 Synchronisierung von Prozessen über Rechner- und Betriebsystemgrenzen hinweg

Unter Windows kann die Systemzeit standardmäßig nur sekundengenau mit Zeitservern synchronisiert werden [6]. Die zur späteren Synchronisation von Film und Folien notwendige Synchronisation der Aufzeichnungsprozesse auf dem Videoaufzeichnungsrechner und dem Dozentenrechner auf wenige Decisekunden kann daher im Regelfall nicht über einen Rückgriff auf die Systemzeit realisiert werden.
Erweiterte Möglichkeiten zur Zeitsynchronisation sind durch so genannte Rechner-Funkuhren gegeben, die als Hardware-Karten im Rechner verbaut werden können [Samulat 2000]. Derartige Rechner-Funkuhren basieren entweder auf dem DCF77 Standard oder verwenden das satellitengestütze Global Positioning System GPS als Quelle der Referenzzeit. Mit DCF77 ist eine Synchronisierung im Millisekundenbereicht möglich, bei GPS beträgt die Abweichungstoleranz sogar nur wenige Mikrosekunden. DCF77 ist jedoch aufgrund der Übertragung der Synchronisationsdaten im Niederfrequenzbereich sehr störanfällig, GPS-basierte Karten benötigen eine externe Außenantenne, um eine Verbindung zum Satelliten herzustellen [Samulat 2000]. Der Anschluss einer solchen Außenantenne an den Dozentenrechner ist im Hörsaalszenario nur schlecht zu realisieren. Hinzu kommt die Tatsache, dass alle verwendeten Dozentenrechner mit der entsprechenden Hardware ausgestattet werden müssten. Ein solcher Schritt würde die Flexibilität der im Rahmen dieser Arbeit entwickelten Aufzeichnungslösung in einem nicht akzeptablen Maße einschränken.
Eine softwareseitige Lösung zur Zeitsynchronisation stellt das Network Time Protocol (NTP) dar. Mit NTP kann die Systemzeit eines Rechners in der Regel auf unter 100 Millisekunden genau synchronisiert werden [Deeths & Bruentte 2001]. Auch für Windows existiert Software, mit der eine Zeitsynchronisation der Systemzeit über NTP möglich ist. Derartige Programme müssen jedoch als Zusatzsoftware auf dem Rechner des Dozenten installiert

werden. Außerdem bedarf die Zeitsynchronisation teilweise mehrerer Minuten bis hin zu Stunden [Deeths & Bruentte 2001]. Gründe hierfür sind beispielsweise die mehrschrittige Arbeitsweise von NTP und die Tatsache, dass alleine die im Rahmen von NTP benötigte Bestimmung der durchschnittlichen Latenzzeit der Netzwerkverbindung in der Regel fünf Minuten in Anspruch nimmt.
Als Fazit dieser Betrachtung lässt sich festhalten, dass die Nutzung von Systemzeiten zur Prozesssynchronisation im Rahmen der erarbeiteten Softwarelösung keinen gangbaren Weg darstellt. Zur Synchronisation von Videoaufzeichnung und Präsentationsprotokoll wird daher mit direkten Synchronisationssignalen zwischen den betreffenden Rechnern gearbeitet.

#### 7.2.1.2 Das Videoformat MPEG-2

Zur Aufzeichnung von Videosignalen existiert eine Vielzahl von unterschiedlichen Videoformaten. Im Rahmen der vorliegenden Arbeit wurde das MPEG-2 Format gewählt, da es eine Reihe von Vorteilen mit sich bringt.
Zum einen sind Hardware-Karten erhältlich, wie beispielsweise die Hauppauge PVR 150 MPEG-Encoder-Karte, mit denen über eine AV-Verbindung eingehende Videosignale direkt in das Format umgewandelt werden können. Auf diese Weise können Codierungsartefakte, die bei frühen Versuchen der Aufzeichnung von Vorlesungen an der Universität Osnabrück mit software-basierten Lösungen wie dem Real Producer Probleme aufgetreten sind, vermieden werden.
Ein weiterer Vorteil des MPEG-2 Formates ist die Möglichkeit, in DVD-Qualität aufzuzeichnen. So können die aufgezeichneten Videos auch mit hoher Qualität wiedergegeben werden, wie beispielsweise beim Einsatz der Aufzeichnung als zeitfix präsentierter Vorlesungsersatz auf großer Leinwand nötig (vgl. 3.2).
Durch nachgelagerte Konvertierung kann das im MPEG-2 Format aufgezeichnete Video später in andere Wiedergabeformate umgewandelt werden. Bei Aufzeichnung mit Hilfe einer entsprechenden Encoder-Karte sind daher sowohl Zuverlässigkeit und Qualität der Aufnahme gesichert als auch Möglichkeiten zur Wiedergabe in unterschiedlichen Formaten gewährleistet.

### 7.2.2 Details der Implementation

Die Aufgaben der Aufzeichnungskomponente lassen sich in vier wesentliche Bereiche einteilen. Einer dieser Bereiche ist die Protokollierung von einzelnen Präsentationsereignissen wie Folienwechseln oder Animationsschritten. Der zweite Hauptbereich ist die Kommunikation zu anderen Rechnern, wie auch in Abb. 7-3 auf Seite 133 zu erkennen ist. Den dritten und vierten Schwerpunkt der Implementation stellen der auf dem MPEG2-Aufzeichnungsrechner installierte Videoaufzeichnungsserver und die Benutzerschnittstelle der Aufzeichnungskomponente dar. Im Folgenden wird zunächst kurz der Ablauf des Aufzeichnungsprozesses beschrieben. Daran anschließend werden die einzelnen Teilbereiche der Implementation separat dargestellt.

#### 7.2.2.1 Ablauf des Aufzeichnungsprozesses

Mit dem Starten einer Präsentation werden zeitgleich zwei verschiedene Prozesse angestoßen. Zum einen wird mit der Abarbeitung des in 7.1 beschriebenen Algorithmus zur Indizierung der Folienanimationen der PowerPointpräsentation begonnen. Zum anderen wird eine Verbindung zum MPEG2-Aufzeichnungsrechner aufgebaut, über die die Aufzeichnung des Videos gestartet wird. Abb. 7-4 gibt einen Überblick über die einzelnen Schritte des Aufzeichnungsprozesses.

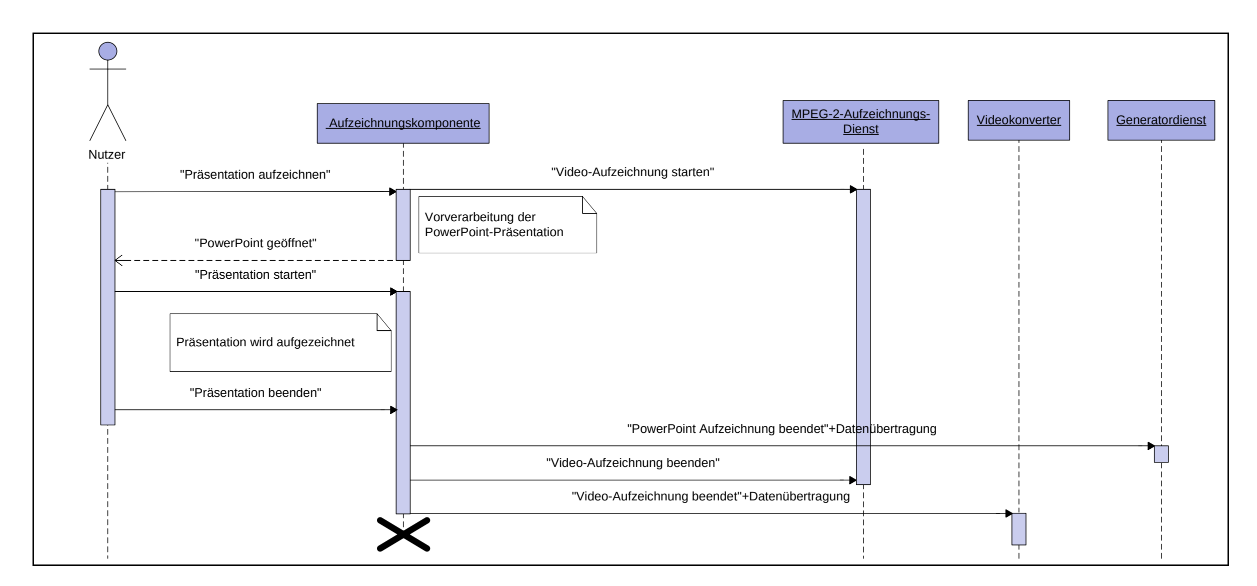



**Abb. 7-4: UML-Sequenzdiagramm zum Ablauf der Aufzeichnung**

Nach Beendigung der Vorverarbeitung der Präsentation wird PowerPoint geöffnet. Anschließend startet der Nutzer die Präsentation. Während der Präsentation werden alle Folienwechsel und Animationen protokolliert. Nach Beenden der Präsentation wird eine Nachricht zum Beenden der Videoaufzeichnung an den MPEG2-Aufzeichnungsrechner gesendet. Mit einer kurzen Zeitverzögerung (vgl. 7.2.2.3) wird eine Nachricht an den Videokonvertierungsserver gesendet und die Protokolldatei der Präsentation wird samt der zugehörigen PowerPoint-Datei an den Konvertierungsserver übertragen. Durch die gesendeten Nachrichten wird die in 7.3 beschriebene Nachbearbeitungsphase angestoßen.

#### 7.2.2.2 Protokollierung von Präsentationsereignissen

Wie bereits in 7.1.1 dargestellt, besteht von C# aus die Möglichkeit mit PowerPoint über die COM-Schnittstelle der Applikation zu kommunizieren. Wie in der Einleitung zu 7.1 erwähnt, kann PowerPoint über diese Schnittstelle nicht nur ferngesteuert werden. PowerPoint verfügt auch über einen Event-Mechanismus, der verschiedene Präsentationsereignisse wie den Übergang in den Präsentationsmodus, das Verlassen des Präsentationsmodus und auch Folienereignisse wie Animationsschritte und Folienwechsel an externe Eventhandler propagiert. In [20] findet sich eine detaillierte Beschreibung dazu, wie diese Event-Schnittstelle von C# aus genutzt werden kann.

Durch diese Event-Schnittstelle konnte die Protokollierung von Folienereignissen ohne komplexere Schritte implementiert werden. Die bereits in 7.1.2 erörterte Problematik des Fehlens einer Möglichkeit zur eindeutigen Identifikation von Folienschritten wird durch die ebenfalls in 7.1.2 beschriebenen Vorbereitungsschritte umgangen. Bei jedem Folienwechsel und jeder Folienanimation wird der betreffende Schritt zusammen mit dem jeweils aktuellen Zeitindex in einer Datei protokolliert.

#### 7.2.2.3 Zeitkritische Rechnerkommunikation und die Windows Firewall

Da die Synchronisation der Videoaufzeichnung und des Präsentationsprotokolls durch direkte Kommunikation per TCP/IP zwischen dem Dozentenrechner und dem MPEG2-Aufzeichnungsrechner realisiert ist (vgl. 7.2.1.1), ist der Austausch von Nachrichten zwischen diesen Rechnern stellenweise zeitkritisch.

Ein Hindernis stellt hier eine Kuriosität der Windows Firewall dar. In einigen Versionen konnte während der Entwicklung und während des Einsatzes der Aufzeichnungskomponente reproduzierbar eine ca. 30 Sekunden andauernde Verzögerung zwischen dem Aufbau einer Socket-Verbindung und der Übertragung des ersten Signals über diese Verbindung

beobachtet werden. Diese Verzögerung tritt auf, obwohl die Applikation in der Firewall als genehmigt markiert ist.
Bei zeitkritischer Kommunikation zum Videoaufzeichnungsserver wird zur Sicherung der Synchronisation daher ein Sicherheitsintervall zwischen dem Herstellen einer Verbindung und der eigentlichen Kommunikation berücksichtigt.

#### 7.2.2.4 Videoaufzeichnungsserver

Der MPEG2-Aufzeichnungsrechner verfügt über eine Hauppauge PVR 150 MPEG-Encoder-Karte und eine Linux-Installation. Unter Linux kann die Karte bei vorheriger Installation der nötigen Hardwaretreiber als Device gemountet werden. Die Karte encodiert eingehende AV-Signale werden direkt und bietet die Möglichkeit, die Signale als Daten im MPEG-2-Format aus dem entsprechenden Device zu lesen. So ist es möglich, die Daten direkt in eine Datei zu schreiben
Der Service selbst ist als Shell-Script implementiert. Bei Verbindungsaufbau zum Videoaufzeichnungsserver wird das Script per *xinetd* gestartet. Die eingehende Socket-Verbindung wird dabei automatisch mit den *stdin*-, *stdout*- und *stderr*-Strömen des Prozesses verbunden, in dem das Script gestartet wird. Daran anschließend ist über die Socket-Verbindung zunächst eine Authentifizierung erforderlich. Nach erfolgreicher Authentifizierung kann eine Aufzeichnung gestartet bzw. eine bereits laufende Aufzeichnung beendet werden. Die Aufzeichnung selbst findet durch die Umleitung der Ausgabe des durch die PVR 150 erzeugten Devices in eine Datei statt. Anhand der Codierungsparameter der Aufzeichnung und des freien Speicherplatzes auf der Festplatte kann darüber hinaus eine Worst-Case-Abschätzung der verbleibenden Aufzeichnungszeit ermittelt werden. Eine genaue Bestimmung dieses Wertes ist nicht möglich, da die Kompressionsrate des MPEG-2-Formates abhängig von den Eingabedaten variiert.
Beim Starten des Auslesens der Hauppauge PVR 150 über das Script treten Verzögerungen im Sekundenbereich auf. Der Start einer Videoaufzeichnung ist daher zur Synchronisation ungeeignet. Das Beenden des durch das Script gestarteten Schreibprozesses in eine Datei ist hingegen hinreichend genau steuerbar. Die Videoaufzeichnung wird daher vor der Aufzeichnung der PowerPoint-Präsentation gestartet und genau mit dem Abbruch der Aufzeichnung beendet. Damit das Ende der Videoaufzeichnung und der PowerPoint-Aufzeichnung zeitlich synchron. Um auch die Anfänge zu synchronisieren, wird in der Nachbearbeitungsphase der Zeitversatz vom Anfang der Videodatei durch Videoschnitt entfernt.
Um die weitere Nutzung der erzeugten MPEG-2-Datei durch nachgelagerte Verarbeitungsprozesse zu erlauben, ist auf dem Videoaufzeichnungsserver ein ftp-Server installiert, der Zugriff auf alle vom Aufzeichnungs-Script erzeugten Dateien ermöglicht.

#### 7.2.2.5 Benutzerschnittstelle der Aufzeichnungskomponente

Die Benutzerschnittstelle der Aufzeichnungskomponente ist auf eine minimale Menge von Funktionalitäten beschränkt. Daten wie die IP-Adressen des verwendeten Aufzeichnungsrechners und der im weiteren Verlauf der Produktionskette anzusteuernden Rechner werden in einer XML-basierten Konfigurationsdatei gespeichert und sind dem Dozenten über die Benutzerschnittstelle nicht zugänglich. In dieser Datei wird auch eine ID abgelegt, mit der die Aufzeichnung später eindeutig der Veranstaltung zugeordnet werden kann.
Wie in Abb. 7-5 dargestellt, zeigt die Eingabemaske der Aufzeichnungskomponente veranstaltungsbezogene Daten wie den Namen des Dozenten und den Titel der Veranstaltung an. Aus einer Drop-Down-Liste können die Adressen verschiedener Videoaufzeichnungsrechner ausgewählt werden. Die Adresse, die in der Konfigurationsdatei angegeben ist, wird dabei standardmäßig aktiviert. Das einzige vom Nutzer obligatorisch zu

füllende Feld beinhaltet den Pfad zur aufzuzeichnenden PowerPoint-Präsentation. Zum Suchen der jeweiligen Präsentation wird ein Datei-Explorer geöffnet.

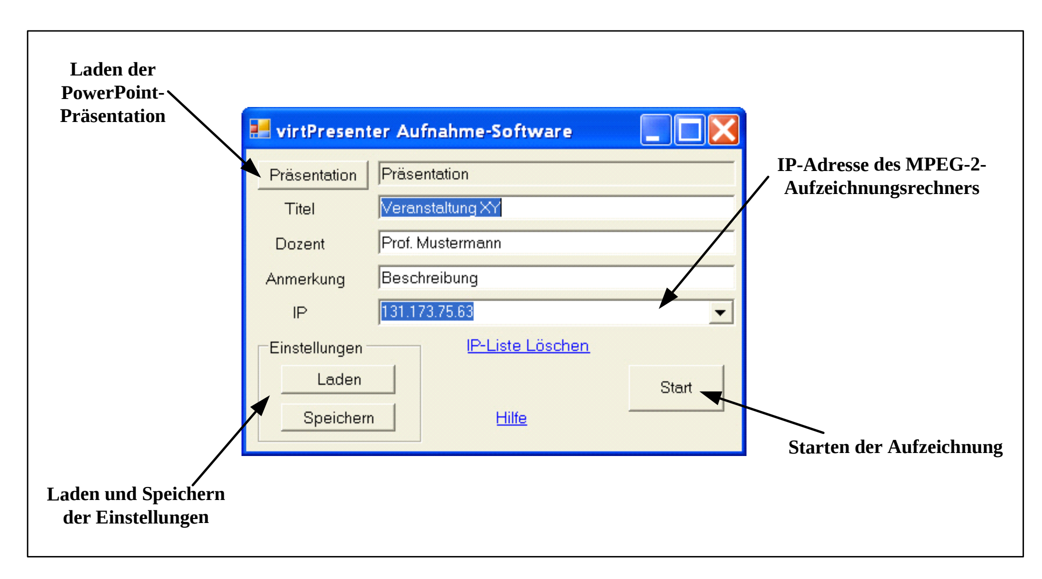



**Abb. 7-5: Eingabemaske der virtPresenter-Aufzeichnungskomponente**

Wenn der Nutzer die Aufzeichnung startet, erscheint zunächst ein Statusbalken der den Fortschritt der Vorverarbeitung und des Aufbaus der Verbindung zum Aufzeichnungsrechner anzeigt. Ist die Videoaufzeichnung gestartet und die Vorverarbeitung der Präsentation abgeschlossen, wird PowerPoint geöffnet. Sobald der Nutzer die Präsentation startet, werden automatisch alle Folienwechsel und Animationsschritte protokolliert (vgl. 7.2.2.2). Bei Beenden der Präsentation erscheint ein Dialog, der es ermöglicht die Präsentation fortzusetzen oder die Aufzeichnung zu beenden.
Wird dieser Dialog vom Nutzer übersehen, läuft die Aufzeichnung weiter, bis die Aufzeichnungskomponente ordnungsgemäß beendet wird. Um den Nutzer auf den Dialog aufmerksam zu machen, ist es sinnvoll, diesen direkt im Vordergrund zu öffnen. Seit der Version 98 ist es in Windows aus einer Applikation heraus jedoch nur dann möglich einen Dialog direkt in den Vordergrund zu bringen, wenn die Applikation im Vordergrund ist [Peterson 1999]. Vor dem Beenden des Präsentationsmodus steht allerdings PowerPoint und nicht die Aufzeichnungssoftware im Vordergrund der Bildschirmanzeige. Da ein Minimieren des PowerPoint-Fensters über die COM-Schnittstelle nicht möglich ist, kann der Dialog nicht direkt aus der Applikation heraus im Vordergrund platziert werden. In C# können jedoch in C geschriebene Funktionen des Betriebsystems als externe Funktionen deklariert und verwendet werden. Wie in Code-Beispiel 7-1 gezeigt, kann so die Funktion `AttachThreadInput` genutzt werden, mit der sowohl Eingabeereignisse geteilt werden können als auch der Fokus zu einer anderen Anwendung transferiert werden kann.

```
...

[DllImport("user32.dll")]
 public  static extern int  SetForegroundWindow(IntPtr hwnd);
[DllImport("user32.dll")]
 public static extern IntPtr AttachThreadInput(IntPtr idAttach, IntPtr
      idAttachTo, int fAttach);
[DllImport("user32.dll")]
 public static extern IntPtr GetWindowThreadProcessId(IntPtr hWnd,
      IntPtr ProcessId);
...

private void windowToFront()
       {
          //get window handle from current process
            IntPtr hwndMain =
               Process.GetCurrentProcess().MainWindowHandle;

           //get thread ID from application currently in foreground
            IntPtr ThreadID1 =
               GetWindowThreadProcessId(GetForegroundWindow(),
               IntPtr.Zero);
           //get thread ID from current process
            IntPtr ThreadID2 =
                GetWindowThreadProcessId(hwndMain, IntPtr.Zero);

            if (ThreadID1 != ThreadID2) //if current thread
               {                        // does not have focus
                       //merge input of threads
                 AttachThreadInput(ThreadID1, ThreadID2, 1);
                  //bring window to front
                 SetForegroundWindow(hwndMain);
                  //separate input again
                 AttachThreadInput(ThreadID1, ThreadID2, 0);
               }
            else
               {
                SetForegroundWindow(hwndMain);
                }

       }
...
```

**Code-Beispiel 7-1: Beispielcode zur Einbindung von Funktionen des Betriebsystems in C#**

Ein weiterer Punkt, bei dem die Logik der verwendeten Softwarekomponenten etwas umgangen werden muss, ist die Protokollierung von Folienwechseln, die mittels direkter Tastatureingabe herbeigeführt werden. Werden in der Präsentationsansicht bei einer animierten Folie Animationen abgespielt, kann jederzeit durch Eingabe der Foliennummer einer anderen Folie zu der betreffenden Foliennummer gewechselt werden. Wird anschließend wieder zu der ursprünglichen Folie zurück gewechselt, wird diese im jeweiligen Animationszustand dargestellt. Bei der Protokollierung von Folienwechseln lässt sich der Animationszustand einer Folie jedoch nicht erfassen. Aus diesem Grund werden die Animationszustände aller Folien von der Aufzeichnungskomponente zentral in einem Array verwaltet und zur Erstellung der Protokolldatei verwendet.

## 7.3 Nachbearbeitungsphase

In der Nachbearbeitungsphase wird eine Reihe von Schritten ausgeführt, die der Umwandlung der aufgezeichneten Daten in eine internetgerechte Form dienen. Das im MPEG-2-Format aufgezeichnete Video wird in das Real Video Format konvertiert und die PowerPoint-Präsentation in das Vektorgrafikformat SVG (vgl. 7.4.1.1) umgewandelt. Anschließend werden die Daten in eine Benutzermaske der in Kapitel 1 dargestellten Form eingebunden und auf einem Webserver zur Verfügung gestellt.
In Abb. 7-6 ist grob der Ablauf der Nachbearbeitungsphase skizziert. Nach Beenden der Aufzeichnung wird zeitgleich das Video im MPEG-2-Format vom MPEG2-Aufnahmerechner auf den Real-Konverter geladen (1) und die PowerPoint-Präsentation wird zusammen mit den protokollierten Daten zu Folienwechsel- und Animationszeiten auf den so genannten Generator-Rechner geladen (2).

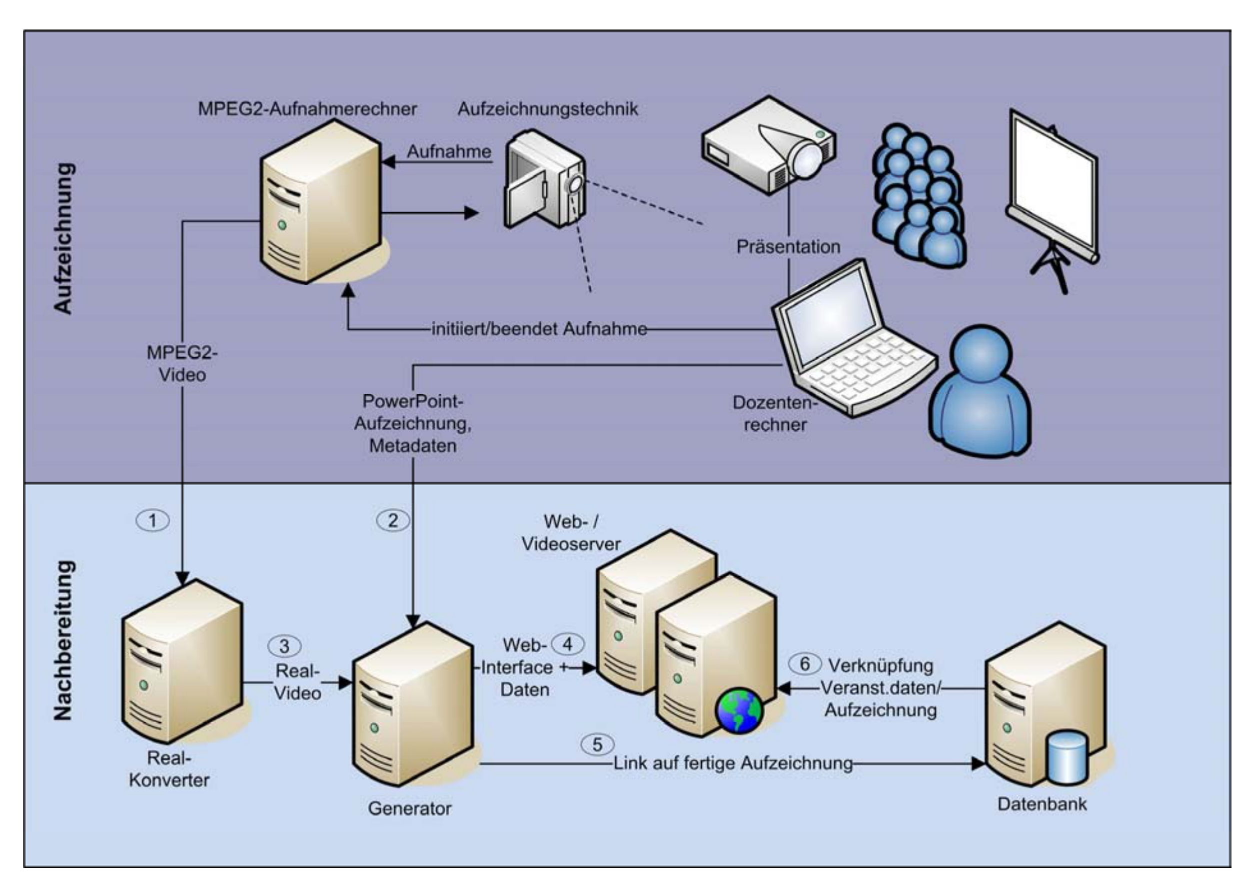



**Abb. 7-6: Verteilung der Nachbearbeitungsphase auf mehrere Rechner**

Der Real-Konverter konvertiert das eingetroffene MPEG-2-Video in eine Real Video Datei, schneidet die Datei zur Synchronisation (vgl. 7.3.2.1) und sendet diese anschließend an den Generator-Rechner (3). Ist das Video auf dem Generator-Rechner eingetroffen, wird auf dem Generator-Rechner die PowerPoint-Präsentation in SVG konvertiert. Ist dieser Vorgang abgeschlossen, werden die SVG-Dateien in ein Web-Interface eingebunden und zusammen mit dem Interface-Code auf einen Web-Server verschoben (4). Die zugehörige Real-Video-Datei wird auf einem streaming-fähigen Video-Server abgelegt (4), die Adresse der Videodatei ist in das Web-Interface eingebunden.[69] Die Adresse des Web-Interfaces der Aufzeichnung wird zusammen mit Metadaten wie einer eindeutigen ID der aufgezeichneten

[69] Die Speicherung des Real-Videos auf dem Generator-Rechner ist zwar mit einem zusätzlichen Übertragungsaufwand im Gegensatz zum direkten Verschieben des Videos auf den Videoserver verbunden, ist jedoch für Backups und bei der Erstellung von offline-Medien hilfreich.

Veranstaltung in einer Datenbank abgelegt (5). Damit ist es möglich, alle Aufzeichnungen eines Kurses zentral zu verwalten und mit den Daten der Veranstaltung zu verknüpfen (6).
Die eigentliche Nachbearbeitung findet dabei auf dem Realkonverter und dem Generator-Rechner statt. Da der Generator PowerPoint-Präsentationen verarbeitet, ist er wie auch die Aufzeichnungs-komponente in C# implementiert. Zur Kommunikation mit PowerPoint wird auch hier die COM-Schnittstelle der Applikation genutzt (vgl. 7.1.1.1). Um die Wiederverwendbarkeit von Code zu ermöglichen, ist auch die auf dem Realkonverter Installierte Software in C# geschrieben. Beide Programme sind als Windows-Dienste implementiert. Sie können daher auf den jeweiligen Rechnern im Hintergrund arbeiten, ohne dass ein Benutzer auf dem Rechner angemeldet sein muss.
Im weiteren Verlauf dieses Abschnittes werden zunächst Grundlagen zur Implementation von Diensten unter Windows, zur Verarbeitung von Videodateien mit der verwendeten Konvertierungssoftware und zur Erstellung von SVG aus PowerPoint behandelt. Daran anschließend werden für den Ablauf der Produktionskette relevante Details der Implementation behandelt. Da eine adäquate Behandlung der zur Konvertierung von PowerPoint zu SVG verwendeten Software den Rahmen dieses Kapitels sprengen würde, findet eine ausführliche Darstellung dieser Software in Kapitel 8 statt.

### 7.3.1 Grundlagen

Im Vergleich zur Anbindung von Programmen an *xinetd* unter Linux (vgl. 7.2.2.4) ist die Entwicklung von Diensten unter Windows komplizierter. Windows-Dienste stellen jedoch die Möglichkeit der Fernsteuerung von Office Applikationen zur Verfügung. Die Implementation von Diensten unter Windows in der Programmiersprache C# stellt daher einen Schwerpunkt dieses Abschnittes dar.
Der zweite Schwerpunkt des Abschnittes liegt auf der Darstellung der zur Konvertierung der Datenformate MPEG-2 und PowerPoint in die webtauglichen Endformate Real Video und SVG nötigen Schritte.

#### 7.3.1.1 Implementation von Windows-Diensten unter C#

Das Grundgerüst eines Windows-Dienstes kann beispielsweise mit der Entwicklungsumgebung Visual Studio erstellt werden. Dazu muss beim Erzeugen eines neuen Projektes die Projektvariante Windows-Dienst ausgewählt werden. Die zu implementierende Dienst-Klasse muss in der Klassenhierarchie von der Klasse `System.ServiceProcess.ServiceBase` abstammen. Damit der erstellte Dienst später auf einem Windows-System installiert werden kann, muss das Projekt manuell mit einem `System.Configuration.Install.Installer` versehen werden.
Schwierigkeiten bei der Implementation von Windows-Diensten sind vor allem bei der Fehlerbehandlung gegeben. Dienste sind auch dann aktiv, wenn kein Benutzer auf dem Rechner eingeloggt ist. Daher ist es nicht möglich, Fehlermeldungen über Messageboxen oder die Konsole anzuzeigen.[70] Es empfiehlt sich aus diesem Grunde, Fehlermeldungen in eine zu diesem Zweck angelegte Datei zu schreiben. Bei Installation des Dienstes auf einem neuen Rechner ist die Existenz des nötigen Dateipfades sicherzustellen. Ein Debuggen von Diensten ist im Visual Studio nicht möglich, da ein Dienst zunächst als solcher installiert werden muss und anschließend über das Betriebsystem verwaltet wird.
Eine weitere Besonderheit bei der Entwicklung von Diensten stellt deren Startverhalten dar. Beim Start des Dienstes ruft das Betriebssystem die Methode `OnStart (string[] args)` auf. Da die Laufzeit dieser Methode zeitlich auf wenige Sekunden begrenzt ist, ist es nicht

[70] Prinzipiell ist es möglich, in den Einstellungen eines Dienstes auch Kommunikation mit dem Desktop zu erlauben (siehe 7.3.1.2). Für den realen Einsatz ist ein solches Vorgehen jedoch ungeeignet.

ratsam, an dieser Stelle zeitaufwändige Aufgaben zu bearbeiten. Der Code wird zwar kompiliert, der Startvorgang des Dienstes wird jedoch nach Überschreiten der Zeitdauer abgebrochen.
Umgangen werden kann diese Beschränkung durch das Starten eines separaten Threads. Programmcode wie die Methode `Accept()` einer Instanz der Klasse `System.Net.Socket` sollte daher in jedem Fall in einen solchen Thread ausgelagert werden.

#### 7.3.1.2 Installation, Start und Konfiguration von Windows-Diensten

Bevor ein Dienst auf einem Windows-System zu Verfügung steht, muss er zunächst durch Aufruf des Kommandos `installutil.exe` installiert werden [11]. Ist bereits eine ältere Version des Dienstes installiert, muss diese zunächst mit dem Schalter `/u` entfernt werden.
Im Anschluss an die Installation kann der Dienst über den Menüpunkt *Systemsteuerung->Verwaltung->Dienste* im Windows-Startmenü aktiviert werden. Dort kann unter den Eigenschaften des Dienstes auch der Starttyp festgelegt werden. In der Regel empfiehlt es sich, einen Dienst automatisch zu starten, damit er mit jedem Hochfahren des Rechners gestartet wird. Darüber hinaus können in diesem Menüpunkt verschiedene weitere Eigenschaften definiert werden. So ist es möglich, dem Dienst an dieser Stelle Startparameter zu übergeben, den Dienst unter speziellen Benutzerkonten anzumelden und den Datenaustausch mit dem Desktop zu unterbinden oder zu erlauben.
Die Konfiguration eines Dienstes kann entweder über dessen Startparameter oder über eine Konfigurationsdatei realisiert werden. Das Ausführungsverzeichnis von Diensten ist das Systemverzeichnis WINDOWS\system32, ist die Datei an einem anderen Ort hinterlegt, muss der entsprechende Pfad entweder direkt im Code oder besser über Startparameter angegeben werden.

#### 7.3.1.3 Erzeugung von SVG aus PowerPoint

Zur Erzeugung von SVG aus PowerPoint wird im Rahmen der vorliegenden Arbeit die kommerzielle Software SVGmaker [41] verwendet. Der SVGmaker wird unter Windows als Druckertreiber installiert und kann die Druckausgabe beliebiger Programme als SVG speichern [Rotard, Schweikhardt & Ertl 2002]. Der SVGmaker ist als so genannter Kernel Modus Druckertreiber implementiert [Phillips & Kidson 2005]. Kernel Modus Treiber arbeiten in der CPU auf Ring 0 und bringen daher den Vorteil mit sich, dass sie direkt auf Hardwarekomponenten zugreifen können. Im Vergleich zu so genannten User Modus Treiber sind sie jedoch auch mit einer Reihe von Nachteilen verbunden. So laufen User Modus Treiber in der Regel stabiler, sind einfacher zu debuggen und verfügen über eine Reihe weiterer Vorteile [14]. Die Entscheidung für die Implementation des SVGmaker als Kernel Modus Treiber ist daher vermutlich eher historisch begründet.
Die Verwendung des SVGmakers zur Erzeugung von SVG aus PowerPoint über C# gestaltet sich vergleichsweise einfach. Über PowerPoints COM-Schnittstelle kann der SVGmaker als aktiver Drucker ausgewählt werden. Anschließend können über die COM-Schnittstelle Druckaufträge an den SVGmaker gesendet werden. Wie bei anderen Druckertreibern auch ist es möglich, entweder die gesamte Präsentation oder eine Auswahl einzelner Folien zu drucken. Eine Einstellung der Druckoptionen ist wie bei anderen Druckertreibern auch aus der Systemsteuerung heraus möglich.
Probleme bei der Erzeugung von SVG mit dem SVGmaker können sich dann ergeben, wenn viele Druckaufträge sehr schnell hintereinander abgesetzt werden. So ist mit dem in 8.1 beschriebenen Algorithmus zur Erzeugung von SVG unter dem Betriebsystem Windows Server 2003 reproduzierbar das Problem aufgetreten, dass durch den SVGmaker erzeugte Dateien falsche Daten enthielten und andere zu erzeugende Dateien nicht erzeugt wurden. Eine Lösung für derartige Probleme wird in 7.3.2.2.3 vorgestellt.

#### 7.3.1.4 Konvertierung von MPEG-2 zu RealVideo

Zur Konvertierung der aufgezeichneten MPEG-2-Dateien in das Real Video Format wird die kommerzielle Software Real Producer [40] verwendet.
In der Standardkonfiguration ist es mit dieser Software nicht möglich, Dateien im MPEG-2-Format zu lesen. Zusätzlich müssen dazu die erforderlichen Codecs installiert werden, die beispielsweise im K-Lite Codec Pack enthalten sind [37].
Zur Steuerung des Real Producers existiert eine COM-Schnittstelle, die jedoch in neueren Versionen nicht mehr unterstützt wird. Zur Steuerung des Real Producers wird daher die Kommandozeilenschnittstelle der Software verwendet (vgl. 7.3.1.7).

#### 7.3.1.5 Ansprechen von Office Applikationen aus Diensten heraus

Prinzipiell basiert die Kommunikation des Generatordienstes mit PowerPoint auf den in 7.1.1.1 beschriebenen Grundlagen. Im Gegensatz zur Aufzeichnungskomponente ist der Generatordienst jedoch nicht als interaktive Applikation sondern als autonomer Dienst implementiert.
Bei der Steuerung von Office Applikationen durch Dienste ist eine Reihe zusätzlicher Faktoren zu beachten [22]. So ist zum Beispiel beim Auftreten von Fehlern in der Regel keine Interaktivität mit dem Desktop möglich. Dieses Faktum ist besonders hinsichtlich der Tatsache, dass einige Fehlermeldungen von Office Applikationen nicht über die COM-Schnittstelle an den aufrufenden Prozess übermittelt werden, sondern direkt einen Nutzerdialog starten, kritisch. Das Auftreten derartiger Fehlermeldungen muss daher durch vorausschauende Programmierung vermieden werden.
Eine weitere Problemquelle stellt die Tatsache dar, dass Office Applikationen bei der Initialisierung auf das Benutzerprofil des aktuellen Nutzers zugreifen. Standardmäßig laufen Dienste unter Windows jedoch unter dem Benutzerkonto System, dass nicht über ein Benutzerprofil verfügt. Daher können beim Start von Office Applikationen aus Diensten heraus Fehler beim Aufruf der COM-Methoden `CreateObject` oder `COCreateInstance` auftreten. Derartige Fehler treten auch bei der Ansteuerung von PowerPoint aus Diensten heraus auf. Eine Möglichkeit dieses Problem zu umgehen, ist die Zuweisung eines geeigneten Benutzerkontos an den Dienst (siehe 7.3.1.2).
Auch sind Office Applikationen nur bedingt multi-threading fähig. Der Zugriff auf eine Office Applikation muss daher durch das steuernde Programm koordiniert werden. Eine weitere Problemquelle ergibt sich in diesem Zusammenhang dadurch, dass Objekte in PowerPoint teilweise nur mit Hilfe der globalen Zwischenablage kopiert werden können. Wird die Zwischenablage von mehreren Threads gleichzeitig genutzt, können Daten vertauscht werden oder verloren gehen. Lösungen stellen hier einerseits die Synchronisation von Zugriffen auf PowerPoint und die Zwischenablage sowie andererseits die Sequentialisierung von unabhängigen Arbeitsschritten dar. Da die Verarbeitung und teilweise bereits das Laden von PowerPoint Präsentationen sehr speicherintensiv sind, bietet sich das parallele Arbeiten mit mehreren Präsentationen jedoch ohnehin nicht an. Im Rahmen der vorliegenden Arbeit wird daher die in 7.3.2.2.1 näher beschriebene sequenzielle Abarbeitungsstrategie verwendet.
Darüber hinaus stellen die Stabilität von Office Applikationen und die Möglichkeit, Makros in Office-Dokumente einzubetten potentielle Stolpersteine dar. Sicherheitsprobleme durch Macros können durch entsprechende Einstellungen in den jeweiligen Office Applikationen vermieden werden. So kann beispielsweise in PowerPoint das Ausführen von Makros über den Menüpunkt *Extras->Makro->Sicherheit* unterbunden werden. Für Stabilitätsprobleme von Office Applikationen existieren im Gegensatz dazu keine derart einfachen Lösungen. Stabilitätsprobleme bei der Verwendung von Office Applikationen aus Diensten heraus sind unter anderem dadurch bedingt, dass Office weder für den Betrieb auf Servern konzipiert,

noch unter derartigen Bedingungen getestet worden ist. Durch die Microsoft Windows Installer (MSI) Technologie werden einige Softwarekomponenten erst dann installiert oder konfiguriert, wenn sie zum ersten Mal in der Applikation angefordert werden. Im Serverbetrieb wird durch diese Vorgehensweise nicht nur die Leistung des Systems beeinträchtigt, es kann auch vorkommen, dass Dialoge zur Bestätigung des Installationsprozesses geöffnet werden, wodurch der jeweilige Serverprozess zum Stillstand kommt. Diesen Problemen kann nur durch eine gründliche Testphase bei der Installation begegnet werden. Zusätzlich empfiehlt sich die Protokollierung von unerwarteten Fehlermeldungen in speziellen Dateien, die das Auffinden von derartigen Problemen erleichtern.
Zusammengefasst lässt sich festhalten, dass viele Probleme, die bei der Arbeit mit Office Applikationen aus Diensten heraus auftreten können durch vergleichsweise einfache Vorgehensweisen vermieden werden können. Probleme, die durch die Konzeption von Office Applikationen als interaktive Komponenten entstehen, lassen sich nicht sicher vermeiden. Vorausschauende Programmierung und gründliches Testen der verwendeten Applikationen mit den zu erwartenden Eingabeformaten sind jedoch ein guter Weg, um Komplikationen an dieser Stelle zu vermeiden.

#### 7.3.1.6 Prozessüberwachung

Eine Besonderheit des .NET Frameworks ist die Möglichkeit, mittels so genannter Windows Performance Counter (`System.Diagnostics.PerformanceCounter`) einen feingranularen Überblick über den Ressourcenverbrauch einzelner Prozesse zu gewinnen [Henkel 2004]. [Robinson, Nagel, Glynn, Skinner, Watson & Evjen 2004] beschreibt exemplarisch und leider sehr kurz gefasst den Einsatz dieser Technologie zur Überwachung von Diensten unter Windows.
Die Nutzung der Klasse `System.Diagnostics.PerformanceCounter` selbst gestaltet sich recht einfach, wie das Beispiel in Code-Beispiel 7-2 illustriert.

```
PerformanceCounter myCounter=new PerformanceCounter(
"Prozess","Prozessorzeit (%)","POWERPNT");

int result=myCounter.NextValue()
```

**Code-Beispiel 7-2: Überwachung der Prozessorauslastung durch einen PowerPoint-Prozess mittels der Klasse `System.Diagnostics.PerformanceCounter` in C#.**

Problematisch ist beim Einsatz der Klasse `PerformanceCounter` allerdings die Tatsache, dass die Parameter beim Konstruktoraufruf mit der Spracheinstellung des Betriebssystems variieren und auch von der Hardwarekonfiguration des verwendeten Rechners abhängen können. Es existiert daher keine verlässliche Dokumentation zu diesem Punkt. Ein kurzer Überblick über die englische Namensgebung der wichtigsten Kategorien (Prozessor, Speicher etc.) und Messdimension (Auslastung prozentual, Auslastung gesamt usw.) ist in [Wille 2004] zu finden. Ein systemspezifischer Überblick kann mit Hilfe der Klassenmethode `GetCategories()` sowie der Objektmethoden `GetInstanceNames()` und `GetCounters(string instanceName)` der Klasse `System.Diagnostics. PerformanceCounter-Category` erzeugt werden. Über die Variable `CounterName` der der zurück gelieferten `PerformanceCounter` können anschließend die Namen der zugehörigen Messdimensionen ermittelt werden.

Im Rahmen der vorliegenden Arbeit werden Performance Counter genutzt, um die Reallaufzeit von Aufrufen zu bestimmen, die Applikationen wie PowerPoint (siehe 7.3.2.2.2) oder den SVGmaker (siehe 7.3.2.2.3) ansteuern.
Die Nutzung von Performance Countern ist nur unter speziellen Nutzerkonten wie Administrationskonten oder Konten, die sich in der so genannten Performance Counters Users Group befinden, möglich [15]. Die zur Konvertierung der PowerPoint-Präsentation verwendete Nachbearbeitungskomponente muss daher unter einem entsprechenden Nutzerkonto installiert werden (vgl. dazu auch 7.3.1.2).

#### 7.3.1.7 Ansprechen externer Prozesse

Nicht alle Applikationen, die in der Nachbearbeitungsphase verwendet werden, verfügen über eine COM-Schnittstelle. Zur Kommunikation mit diesen Applikationen werden deren Kommandozeileninterfaces verwendet. So wird die zu erzeugende Webmaske auf einem Apache Tomcat Webserver vorgehalten (vgl. 7.4.2.2). Zum Upload und zur Aktivierung einer Webapplikation bietet sich das XML-basierte Softwarewerkzeug ANT an [4]. Da ANT zwar eine API bietet, diese jedoch nur für Java zur Verfügung steht, wird ANT im Rahmen der vorliegenden Arbeit mit Hilfe der Klasse `System.Diagnostics.Process` als externer Prozess gestartet. Auch zur Steuerung des Helix Producers wird ein externer Prozess gestartet (vgl. 7.3.1.4).

```
...

System.Diagnostics.ProcessStartInfo startInfo = new
  System.Diagnostics.ProcessStartInfo("ant.bat");

startInfo.Arguments = "deploy -buildfile"+
                       resultMaskDir+"\\build.xml";
startInfo.WorkingDirectory = resultMaskDir;
startInfo.UseShellExecute = false;//has to be set to false,
                  // otherwise redirecting IO does not work
startInfo.RedirectStandardError=true; // redirecting output to
startInfo.RedirectStandardOutput = true; // caller (this process)

System.Diagnostics.Process p = new System.Diagnostics.Process();
p.StartInfo = startInfo;

p.Start();
string ANT_Error=””, ANT_Out=””;

while (!p.StandardOutput.EndOfStream)
          ANT_Out+=p.StandardOutput.ReadLine();

while (!p.StandardError.EndOfStream)
       ANT_Error+=p.StandardError.ReadLine();
...
```

**Code-Beispiel 7-3: Ansprechen externer Prozesse mit Hilfe der Klasse `System.Diagnostics.Process`**

Wie in Code-Beispiel 7-3 gezeigt, können die Ausgabeströme *stdout* und *stderr* eines mit Hilfe der Klasse `System.Diagnostics.Process` gestarteten Prozesses im aufrufenden Programmcode verarbeitet werden. Auf diese Weise kann auf Fehlermeldungen reagiert werden und Rückgabewerte der jeweiligen Prozesse können weiterverarbeitet werden.
Ein wichtiges Hilfsmittel bei der Arbeit mit der Klasse `System.Diagnostics.Process` ist die Klasse `System.Diagnostics.ProcessStartInfo`. Über die Startinformation können

verschiedene Parameter zur Ausführung des jeweiligen externen Prozesses gesetzt werden. So kann zum Beispiel das Ausführungsverzeichnis des Prozesses gesetzt werden und es kann spezifiziert werden, ob die Ausgabeströme an den aufrufenden Programmcode weitergeleitet werden.

## 7.3.2 Details der Implementation

Wie bereits in der Einleitung zu 7.3 dargestellt, ist die Nachbearbeitungsphase in zwei wesentliche Schritte unterteilt. Zunächst wird die im MPEG-2-Format vorliegende Videodatei konvertiert. Erst, wenn die durch diese Konvertierung erzeugte Real-Video-Datei auf dem Generatorrechner vorliegt (vgl. Abb. 7-6 auf Seite 140), beginnt der zweite Prozessschritt der Nachbearbeitungsphase. Abb. 7-7 schlüsselt die einzelnen Teilaktionen des zweiten Prozessschrittes detaillierter auf.

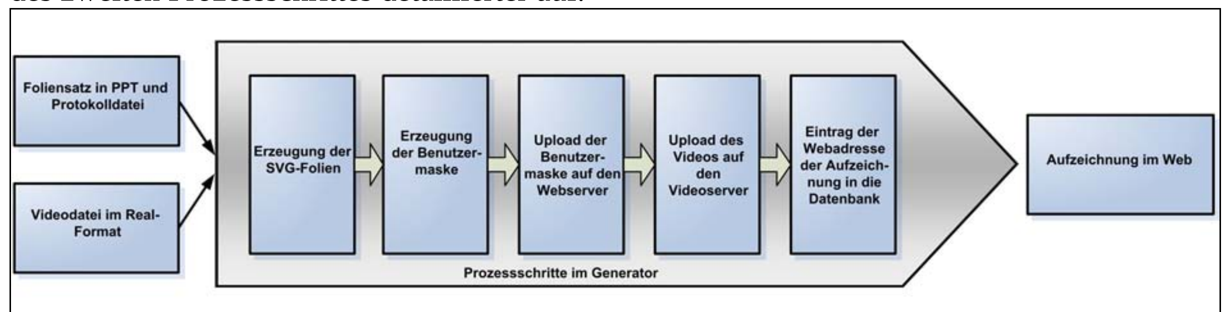



**Abb. 7-7: Schematische Darstellung der Nachbearbeitungsphase**

Zunächst werden aus der verwendeten PowerPoint Datei SVG-Folien erzeugt. In einem zweiten Teilschritt wird unter Einbindung der SVG-Folien die Benutzermaske erzeugt. Anschließend werden die weiteren, bereits in der Einleitung zu 7.3 erörterten, Teilschritte durchgeführt.

Im weiteren Verlauf dieses Abschnittes wird zunächst kurz die zur Konvertierung und zum Schnitt der Videodateien (vgl. 7.2.1.2) implementierte Softwarekomponente dargestellt. Anschließend wird die Implementation des Generatordienstes diskutiert. Die Erzeugung der SVG-Folien wird aufgrund der Komplexität der dazu nötigen Schritte in Kapitel 8 gesondert behandelt.

### 7.3.2.1 Konvertierung und Schnitt der Videodateien

Nach Beendigung der Aufzeichnung sendet der Dozentenrechner ein Signal an den Videokonvertierungsrechner und meldet einen neuen Konvertierungs- und Schneideauftrag. Der Auftrag enthält die Adresse des Rechners, auf dem die zu konvertierende MPEG-2-Datei liegt, die genaue Bezeichnung der Datei, Schnittinformationen der Videodatei und die Adresse des Generatorrechners, an den das Resultat gesendet werden muss.

Eingehende Aufträge werden vom Videokonvertierungsdienst in einer speziellen Queue verwaltet. Die Queue wird nach jeder Aktualisierung serialisiert und bei einem Neustart des Dienstes aus dem Dateisystem geladen. Durch dieses Vorgehen wird Datenverlust bei Beenden des Dienstes, beispielsweise durch eine Sicherheitsaktualisierung des Betriebsystems mit notwendigem Neustart, vermieden. Aktualisierungen der Queue werden beim Eingang eines neuen Auftrages und beim Beenden eines Auftrages vorgenommen. Die Aufträge werden nacheinander in separaten Threads abgearbeitet, um auch bei Abarbeitung eines Auftrages die Annahme weiterer Aufträge zu ermöglichen.

Zur Abarbeitung eines Auftrags wird zunächst die Videodatei per FTP vom MPEG-2-Aufzeichnungsrechner geladen. Als Teil des Videokonverterdienstes ist dazu ein

rudimentärer FTP-Client implementiert.[71] Sobald die Datei auf dem Videokonvertierungsserver eingetroffen ist, wird die Konvertierung in das Real Format mit dem Real Producer als externer Prozess gestartet. Anschließend wird über die COM-Schnittstelle des Real Producers die Ist-Länge des konvertierten Videos bestimmt. Verfügt die zur Konvertierung des Videos verwendete Version des Real Producers nicht über eine COM-Schnittstelle (siehe 7.3.1.4), kann zu diesem Zweck zusätzlich eine ältere Version des Real Producers in der kostenfreien Basic Variante installiert werden. Ist die Ist-Länge bestimmt, wird die Differenz zwischen Soll-Länge und Ist-Länge der Videodatei mit Hilfe des Real Producers entfernt. Aufgrund der Tatsache, dass das Ende der Videoaufzeichnung im Vergleich zu deren Startpunkt hinreichend genau mit dem Ende der PowerPoint-Protokollierung übereinstimmt (vgl. 7.2.2.4), wird die ermittelte Differenz am Anfang des Videos entfernt. Zum Abschluss des Auftrags wird die erzeugte Real-Video-Datei per FTP auf den in der Auftragsbeschreibung angegebenen Generatorrechner geladen und es wird ein Signal an den dort laufenden Generatordienst gesendet.

#### 7.3.2.2 Implementation des Generatordienstes

Die einzelnen Prozessschritte der in Abb. 7-7 auf Seite 146 dargestellten Arbeitsweise des Generatordienstes bedürfen, abgesehen von der in Kapitel 8 behandelten Erstellung und Aufbereitung der SVG-Folien, im Rahmen der vorliegenden Arbeit keiner eingehenderen Darstellung. Der Upload und Aktivierung der Benutzermaske auf den Webserver ist über das Werkzeug ANT als externer Prozess realisiert (siehe 7.3.1.7), für den Datenbankzugriff werden entsprechende Treiber verwendet. Zur Erzeugung der Benutzermaske wird ein Templatemechanismus genutzt, bei dem es ausreicht, die erzeugten SVG-Folien sowie eine Steuerungsdatei mit aufzeichnungsspezifischen Daten wie Folientiteln und Folienwechselzeiten in einen ausgezeichneten Ordner zu kopieren.
Der weitere Verlauf des aktuellen Abschnittes stellt spezifische Probleme, die bei der Realisierung des Generatordienstes aufgetreten sind sowie die implementierten Lösungsansätze vor. Zunächst wird in 7.3.2.2.1 die Verwaltung von eingehenden Aufträgen im Generatordienst diskutiert. Anschließend werden in 7.3.2.2.2 und 7.3.2.2.3 Lösungen diskutiert, mit denen Probleme bei der Überlastung der von PowerPoint bereitgestellten COM-Schnittstelle und die in 7.3.1.3 beschriebenen Probleme bei der programmgesteuerten Erzeugung von SVG aus PowerPoint umgangen werden können.

##### 7.3.2.2.1 Auftragsverwaltung im Generator-Dienst

Da der Generatordienst sowohl das konvertierte und geschnittene Real-Video als auch die auf dem Dozentenrechner protokollierten Daten samt der verwendeten Präsentationsdatei als Eingabe verwendet, kann er erst gestartet werden, nachdem alle zugehörigen Dateien auf den verwendeten Server geladen worden sind (vgl. Abb. 7-7 auf Seite 146). Um festzustellen, wann diese Bedingung eingetreten ist, senden sowohl der Dozentenrechner als auch der Videokonvertierungs-Dienst nach erfolgter Konvertierung bzw. nach erfolgtem Upload der benötigten Dateien auf den Server ein Signal an den Generatorserver. Beide Signale enthalten den Startzeitpunkt der Aufnahme und die IP-Adresse des Video-Aufzeichnungsrechners, um einander eindeutig zugeordnet werden zu können.
Zum Empfang und zur Verarbeitung der Signale ist auf dem Geratorserver der Generatordienst installiert. Der Generatordienst ist ein autonomer Dienst, der automatisch mit dem Hochfahren des Rechners gestartet wird (vgl. 7.3.1.2) und eingegangene Signale in zwei Hashtabellen verwaltet. Sobald die zu einer Protokolldatei passende Videodatei auf den Generatorserver geladen wurde (vgl. 7.3.2.1), wird ein neuer Auftrag zur Verarbeitung der

[71] Eine Beschreibung des File Transfer Protocols FTP findet sich in [34].

Dateien in eine Queue auf dem Generatorserver eingereiht. Abb. 7-8 stellt den zeitlichen Ablauf der Verarbeitung eingehender Signale schematisch in einem Sequenzdiagramm dar. Aus Platzgründen wird die Aufzeichnungskomponente in der Abbildung mit PPT-Listener bezeichnet.

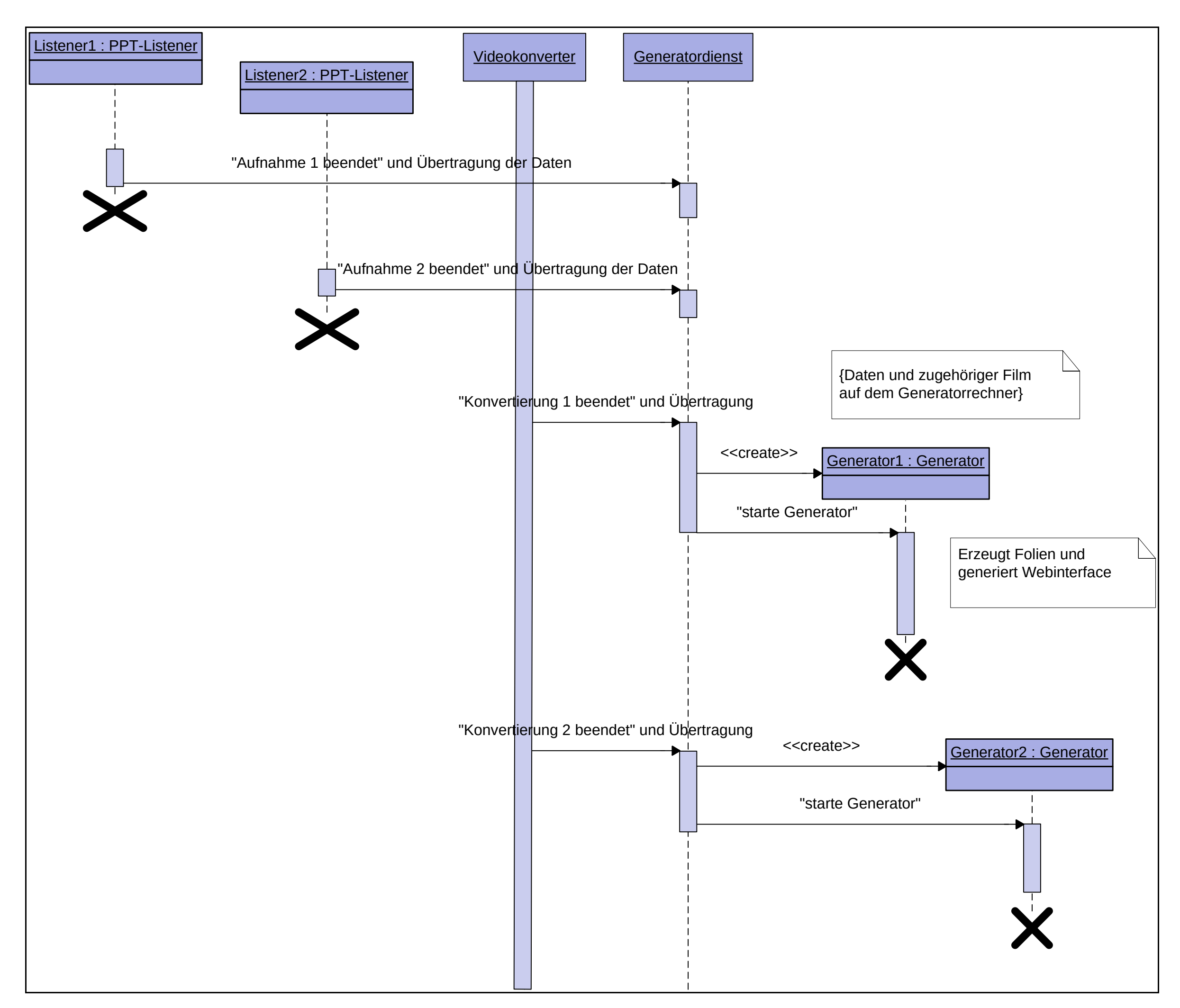



**Abb. 7-8: UML-Sequenzdiagramm zur Abarbeitung eingehender Signale im Generatordienst**

Zum Empfang der Signale arbeitet der Generatordienst mit Socketverbindungen auf zwei unterschiedlichen Ports für PowerPoint-Listener und Videokonverter. Um den Empfang parallel eingehender Signale und auch den Empfang eingehender Signale bei Abarbeitung der Queue zu gewährleisten, wird jeder der beiden Ports von einem separaten Thread kontrolliert.

Bei jedem eingehenden Signal wird geprüft, ob das gleichnamige Gegenstück bereits vorhanden ist und in diesem Fall ein neuer Auftrag in die Queue eingereiht. Bei der Abarbeitung des Auftrages werden Film, Folien und Protokolldateien zur Erstellung einer Webpräsentation genutzt.

Auch die Verwaltung der Queue findet in einem separaten Thread statt und zur Abarbeitung von Aufträgen aus der Queue durch den Generator wird jeweils ein eigener Thread gestartet. Eine parallele Abarbeitung mehrerer Aufträge ist nicht möglich, da im Generator die Zwischenablage des Betriebssystems genutzt wird und PowerPoint nur sehr bedingt multi-threading-fähig ist (vgl. 7.3.1.5). Ein zusätzliches Problem wäre in diesem Kontext die in 7.3.2.2.2 beschriebene Problematik der Überlastung des PowerPoint-Automation-Servers.

Um ein temporäres Blockieren der beiden Kommunikationsports und dadurch bedingte Timeouts bei Kommunikationsversuchen seitens einer Instanz der Aufzeichnungskomponente oder des Videokonverters auszuschließen, wird zur Verarbeitung eines eingehenden Signals nach dem Thread-per-Message-Prinzip [Lea 1999] jeweils ein temporärer Thread erzeugt. Zur Abfrage der Ports wird jeweils ein Objekt der im Rahmen der vorliegenden Arbeit implementierten Klasse `AcceptThread` verwendet, das den jeweiligen Port in einem eigenen Thread permanent überwacht. Dem Konstruktor der Klasse wird der jeweilige Port und ein Factory-Objekt [Gamma, Helm, Johnson & Vlissides 1995] übergeben. Durch die Verwendung des Factory Design-Patterns an dieser Stelle kann die Klasse `AcceptThread` sowohl zum Empfang eingehender Signale vom Videokonverter als auch vom PowerPoint-Listener verwendet werden. Dazu existieren zwei verschiedene Factory-Klassen, die spezielle Worker-Threads erzeugen, die wiederum auf die jeweiligen Aufgaben zugeschnitten sind und eine gemeinsame Schnittstelle in Form einer gemeinsamen Oberklasse implementieren. Aus der Klasse `AcceptThread` heraus wird diese Schnittstelle unabhängig von der jeweils zu erledigenden Aufgabe genutzt.
Bei der Annahme eingehender Signale und bei der Fertigstellung von Aufträgen aus der Queue wird der Zustand der Queue und der oben beschriebenen Hashtabellen für eingehende Nachrichten in einer Datei serialisiert. Wird der Dienst, beispielsweise nach einem Herunterfahren des Rechners durch Stromausfall, neu gestartet, wird der Zustand von Queue und Hashtabellen aus der Datei deserialisiert.
Zur Synchronisation der verwendeten Threads werden die Queue und die zur Verwaltung eingehender Signale des Videokonverters verwendete Hashtabelle genutzt. Da zur Verarbeitung eingehender Signale des PowerPoint-Listeners immer auch der jeweilige Stand eingehender Signale des Videokonverters berücksichtigt werden muss, ist eine Verwendung weiterer Objekte zur Synchronisation nicht sinnvoll. Würde auf beide Hashtabellen synchronisiert werden, müsste eine Synchronisationsreihenfolge festgelegt werden, um Deadlocks zu vermeiden.

##### 7.3.2.2.2 Überlastung des PowerPoint-Automation-Servers

Im Gegensatz zur Steuerung von Office Applikationen über VBA wird bei deren Steuerung über COM die Applikation in einem zweiten Prozess ausgeführt (siehe 7.1.1.1). Daher können unerwartete Effekte beispielsweise bei der Synchronisation und Abarbeitungsgeschwindigkeit von Aufrufen [12] auftreten. Die Tatsache, dass derartige Probleme auch bei Verwendung von nur einem Client und einem Automation-Server auftreten, ist in der entsprechenden Literatur nicht dokumentiert, was auch dadurch begründet sein kann, dass die Dokumentation hier nur sehr vage und teilweise mehrdeutig ist. Bei der Ausführung des in Kapitel 8 beschriebenen Zerlegungsalgorithmus ohne Maßnahmen zur Gewährleistung einer sequentiellen Abarbeitung von Aufrufen sind reproduzierbar Überlastungen des PowerPoint-Automation-Servers aufgetreten. Grund für diese Überlastungen ist, dass Methodenaufrufe aus dem Programm heraus über die COM-Schnittstelle an PowerPoint bereits dann als abgearbeitet gelten, wenn der Aufruf an den PowerPoint-Prozess gesendet wurde.[72] Dies hat zur Folge, dass der Programmzähler bereits zum nächsten Aufruf springt, unabhängig davon, ob der PowerPoint-Prozess die jeweilige Anfrage bereits beendet hat oder nicht. Werden ohne Unterbrechung, beispielsweise durch zeitaufwendige Berechnungen, viele Anfragen an den PowerPoint-Prozess gerichtet, werden die Aufrufe vom PowerPoint-Automation-Server gesammelt, ohne abgearbeitet zu werden. Durch die so entstandene Überlastung kommt der PowerPoint-Prozess ab einer gewissen Auslastung fast zum Stillstand. An diesem Punkt blockiert der Automation-Server die Annahme von Aufträgen, so dass auch entsprechende

[72] Details der zugrunde liegenden Mechanismen sind in 7.1.1.1 beschrieben.

Methodenaufrufe über die COM-Schnittstelle blockieren. Sobald ein Auftrag durch PowerPoint abgearbeitet wurde, nimmt der Automation-Server wieder einen neuen Auftrag an. Ist eine Überlastung einmal aufgetreten, bewegt sich PowerPoint daher bis zur Beendigung des Prozesses an seiner Leistungsgrenze.
Darüber hinaus können durch die Überlastung bedingt auch Fehler im PowerPoint-Prozess auftreten, die nicht über den COM-Mechanismus als Exceptions verarbeitet werden, sondern als Fehler der Applikation PowerPoint gewertet werden. Die zugehörige Fehlermeldung erscheint in diesem Falle als PowerPoint-eigene Fehlermeldung in Form einer Messagebox auf dem Bildschirm und der PowerPoint-Prozess wird automatisch beendet. Da in diesem Fall keine Rückmeldung zum aufrufenden Programm erfolgt, läuft das Programm zunächst weiter. Der genaue Zeitpunkt des Absturzes lässt sich daher nicht feststellen. Informationen über den Zustand der PowerPoint-Präsentation vor dem Absturz des Programms gehen damit verloren. Aus diesem Grund kann auch der Stand der Abarbeitung des Zerlegungsalgorithmus' nicht rekonstruiert werden.
Um die dadurch bedingten Probleme zu vermeiden, wird die in 7.3.1.6 beschriebene Prozessüberwachung genutzt, um das Programm vor entsprechenden Methodenaufrufen in einer Warteschleife zu halten. Dabei wird der PowerPoint-Prozess überwacht und so lange gewartet, bis die Prozessorauslastung durch den Prozess unter einen vordefinierten Schwellwert fällt. Durch Verwendung der Methode `System.Threading.Thread.Sleep` wird eine Auslastung des Prozessors umgangen. Die frei werdende Prozessorzeit steht damit dem PowerPoint-Prozess zur Verfügung.

#### 7.3.2.2.3 Probleme mit dem SVGmaker bei rascher Abfolge von Druckaufträgen

Die in 7.3.1.3 beschriebenen Probleme bei der Erzeugung von SVG aus PowerPoint deuten auf eine mangelhafte oder fehlende Synchronisation bei der Abarbeitung von Druckaufträgen hin.
Analog zu der Prozessüberwachung, die zur Vermeidung der Überlastung des PowerPoint-Automation-Servers verwendet wird (vgl. 7.3.2.2.2), wird daher eine Prozessüberwachung für Druckaufträge eingesetzt. Da der SVGmaker nicht als separater Prozess arbeitet, wird das Verwaltungsprogramm der Druckerwarteschlange `spoolsv.exe` überwacht. Dabei wird nach dem Absenden eines Druckauftrages mit dem Absenden eines weiteren Druckauftrages so lange gewartet, bis der erste Druckauftrag abgeschlossen, d.h. die Prozessorauslastung durch `spoolsv.exe` unter einen definierten Schwellwert gefallen ist. Durch diese Maßnahme konnten die beschriebenen Probleme beseitigt werden. Es ist daher anzunehmen, dass die zugrunde liegenden Fehlfunktionen tatsächlich aus mangelhafter oder fehlender Synchronisation bei der Abarbeitung von Druckaufträgen durch den SVGmaker resultieren.

## 7.4 Nutzungsphase

In der Nutzungsphase werden die erstellten Aufzeichnungen über das in Kapitel 1 beschriebene Interface auf einem Webserver bereitgestellt. Darüber hinaus existiert die Möglichkeit, Aufzeichnungen als Offline-Medium zu erstellen. Das Offline-Medium ist weitestgehend mit der in diesem Abschnitt vorgestellten Software identisch, bietet jedoch naturgemäß keine Möglichkeit zur Kommunikation mit einem zentralen Server. Daher verfügt es nicht über eine Benutzerverwaltung und bietet keine Möglichkeit zur sozialen Navigation.
Der vorliegende Abschnitt beschreibt die Implementation der online auf einem Webserver bereitgestellten Variante der Aufzeichnung und der verwendeten Infrastruktur. Dabei werden zunächst Grundlagen zu den verwendeten Programmiersprachen und Formaten und anschließend Details der Implementation diskutiert.

### 7.4.1 Grundlagen

Wie aus Kapitel 1 ersichtlich, ist die Umsetzung des im Rahmen dieser Arbeit vorgestellten hypermedialen Navigationskonzeptes für Vorlesungsaufzeichnungen mit einem hohen Maß an Interaktivität in der Benutzeroberfläche verbunden. Dieses in einer Web-basierten Wiedergabemaske zu realisieren, ist nur mit Hilfe geeigneter Medienformate zur Wiedergabe von Film und Folien und mit einer Möglichkeit, die Wiedergabe der einzelnen Medien zu koordinieren und zu synchronisieren, machbar.
Dieser Abschnitt stellt die verwendeten Medienformate SVG und Real Video vor. Darüber hinaus werden im vorliegenden Abschnitt Besonderheiten der Scriptsprache JavaScript, die zur Realisierung interaktiver Funktionalität verwendet wird, erläutert und es werden verschiedene Ansätze zur Synchronisation von Film und Folien diskutiert. Abschließend wird ein kurzer Überblick über JavaServer Pages (JSP) gegeben, die im Kontext der vorliegenden Arbeit dazu genutzt werden, die Benutzerverwaltung der Webschnittstelle zu realisieren.

#### 7.4.1.1 SVG

Die Abkürzung SVG steht für *Scalable Vector Graphics* und bezeichnet ein vektorbasiertes Grafikformat zur Beschreibung von zweidimensionalen Grafiken [46]. Im Rahmen der vorliegenden Arbeit wurde SVG als Medienformat zur Darstellung der aufgezeichneten Folienpräsentation gewählt, weil dargestellte Inhalte beliebig skalierbar sind, das Format ein hohes Maß an Interaktivität ermöglicht und SVG-Dateien nachträglich programmgesteuert editierbar sind. Viele der Eigenschaften von SVG liegen in der Tatsache begründet, dass das Format die vektorbasierte Darstellung von graphischen Inhalten erlaubt. In diesem Abschnitt werden daher zunächst die Potentiale beim Einsatz von vektorbasierten Grafikformaten in Vorlesungsaufzeichnungen diskutiert. Daran anschließend werden die Unterschiede zwischen vektor- und pixelbasierten Grafikformaten dargestellt. Abschließend wird ein kurzer Überblick über die Unterschiede der Vektorgrafikformate SVG und Flash gegeben. Weitere Details zu SVG werden in 8.4 behandelt.

##### 7.4.1.1.1 Potentiale des Einsatzes von Vektorgrafik in Vorlesungsaufzeichnungen

Es existieren verschiedene proprietäre und freie Vektorgrafikformate [Gulbransen, Bartlett, Bingham, Kachur, Rawlings & Watt 2002]. In Systemen zur Wiedergabe von Vorlesungsaufzeichnungen wird Vektorgrafik nur vergleichsweise selten eingesetzt. So wird das Format Flash in den Systemen AOF und Lecturnity [Hürst, Müller & Ottmann 2004], PRISM [Cuthbert, Kubinec, Tanic, Ieong, Wei & Schlossberg 2005] und einem in [Wierzbicki 2003] beschriebenen System eingesetzt. SVG wird in einem in [Lecolinet, Faure, Demeure, Moissinac & Pook 2005] dargestellten System zur Aufzeichnung von Vorlesungen genutzt. Im Zusammenhang mit AOF wurde mit der Darstellung der Folien in SVG experimentiert. Das Format wurde dort jedoch verworfen, da die Darstellung des kompletten Vortrags in einer einzigen SVG-Datei aufgrund der Komplexität des resultierenden SVG-Dokumentes nicht realisierbar ist [Kaindl 2002].
In diesen Ansätzen werden Vektorgrafikformate jedoch nur als reine Darstellungsformate genutzt. Dabei bieten Vektorgrafikformate neben hoher Darstellungsqualität eine Reihe weiterer Vorteile wie die Möglichkeit einzelne Folienbestandteile zu animieren oder mit interaktiven Elementen zu versehen. Auch sind Vektorgrafiken verlustfrei skalierbar. Daher können die in 4.5.3 erarbeiteten Anforderungen nach Skalierbarkeit verschiedener Bestandteile des Interfaces, Visualisierung zeitlicher und struktureller Kontexte sowie eine Verbindung von Zeit und Struktur in der in Kapitel 1 beschriebenen Form mit Vektorgrafikformaten vergleichsweise einfach umgesetzt werden. In SVG wird darüber hinaus Text als

Text repräsentiert.[73] Das Format entspricht damit einer in [Lauer & Ottmann 2002] dargestellten Forderung nach symbolischer Repräsentation von Aufzeichnungsinhalten, die so beispielsweise für Volltextsuche genutzt werden können. Ein systematischer Vergleich zwischen pixel- und vektorbasierten Grafikformaten wird im folgenden Abschnitt gegeben.

#### 7.4.1.1.2 Vektorbasierte und pixelbasierte Grafikformate

Die Besonderheit von Vektorgrafik im Gegensatz zu pixelbasierten Grafikformaten ist, dass **Formen in der Grafik anhand von Zeicheninstruktionen beschrieben** werden [Gulbransen, Bartlett, Bingham, Kachur, Rawlings & Watt 2002]. In pixelbasierten Formaten sind einzelne Formen nicht explizit beschrieben. Die Beschreibung des Gesamtbildes besteht aus einem Raster, in dem einzelnen Regionen des Bildes Farbwerte zugeordnet sind. Damit sind überlappende Bildobjekte und der Hintergrund des Bildes nicht voneinander getrennt und das Bild kann nur noch als Ganzes weiter verarbeitet werden [Teague & Campbell 2003]. Wird ein Objekt im Bild verschoben, ist der anfänglich durch das Objekt überdeckte Hintergrund damit nicht mehr rekonstruierbar. In Vektorgrafikformaten werden geometrische Objekte wie Linien, Kurven oder auch Buchstaben symbolisch mit Hilfe von Attributen wie Anfangspunkten, Endpunkten Liniendicke, Radius (bei Kreisen), Stützpunkte (bei interpolierten Kurven) usw. repräsentiert. Die Objekte und deren Attribute bleiben in der Regel auch dann erhalten, wenn die Objekte von anderen Objekten überlagert sind.[74]
Ein weiterer Unterschied zwischen pixel- und vektorbasierten Grafikformaten wird bei der Skalierung von Bildern deutlich. In vektorbasierten Grafikformaten ist die **verlustfreie Skalierung** von geometrischen Objekten möglich [Teague & Campbell 2003]. In pixelbasierten Grafikformaten wie BMP kommt es dagegen bei der Vergrößerung von Bildausschnitten zu Stufeneffekten, die durch die Aufteilung des Darstellungsrasters bedingt sind. Hinzu kommt, dass pixelbasierte Formate wie JPEG verlustbehaftete Transformationen zur Komprimierung vornehmen, die auf die menschliche Wahrnehmung des Bildes in der jeweils verwendeten Auflösung zugeschnitten sind [Miano 1999]. In der Vergrößerung können daher vor allem bei klar definierten geometrischen Formen offensichtliche Verwaschungen auftreten. Ein Vergleich der Grafikformate BMP, JPEG und SVG bei starker Vergrößerung wird in Abb. 7-9 gegeben.




**Abb. 7-9: SVG, JPEG und BMP bei starker Vergrößerung im Vergleich**

Ein Nachteil von vektorbasierten Grafikformaten zeigt sich bei der Darstellung von Grafiken, die nicht aus klar definierten geometrischen Formen wie Kurven oder Linien bestehen. In diese Kategorie fallen Grafiken, die ursprünglich in pixelbasierten Formaten vorliegen. Problematisch sind dabei vor allem Fotografien [Teague & Campbell 2003]. Es existieren zwar Programme wie beispielsweise Adobe Illustrator [26] oder Inkscape [35], mit denen Rastergrafiken durch so genanntes *Tracing* vektorisiert werden können. Derartig erstellte Vektorgrafiken wirken je nach Erstellungsaufwand jedoch oft unrealistisch oder sind mit übermäßig großen Datenmengen verbunden (siehe Abb. 7-10). In Vektorgrafik-

[73] In Flash wird Text im Gegensatz dazu als Grafik eingebunden [Kunze 2006].
[74] Ausnahmen sind möglich, wenn zwei Objekte von einem Editierungsprogramm zu einem Objekt verknüpft werden.

formaten wie Flash oder SVG wird diese Problematik durch die Möglichkeit umgangen, **pixelbasierte Grafiken einbinden** zu können.

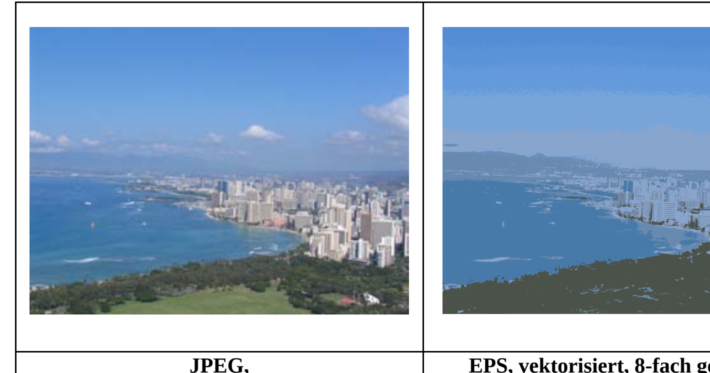
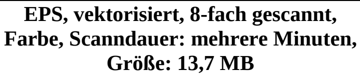
|  | |  |
|---|---|---|
| **JPEG,<br>Auflösung: 1600x1200,<br>Größe: 806 KB** |  **EPS, vektorisiert, 8-fach gescannt,<br>Farbe, Scanndauer: mehrere Minuten,<br>Größe: 13,7 MB** | **EPS, vektorisiert, 72-fach gescannt,<br>Farbe, Scanndauer: mehrere Stunden,<br>Größe: 20,5 MB** |

**Abb. 7-10: Fotografie als JPEG Pixelgrafik und EPS Vektorgrafik im Vergleich.**

In Bezug auf den zur Beschreibung einer Grafik benötigten Speicherplatz kann sich die Verwendung von Vektorgrafikformaten dann als vorteilhaft erweisen, wenn die Grafik überwiegend aus geometrischen Formen wie Linien oder Kurven besteht. Aufgrund der sparsameren Beschreibungsweise im Vergleich zu pixelbasierten Formaten können Grafiken in diesem Fall in Dateien mit einer deutlich geringeren **Dateigröße** gespeichert werden [Teague & Campbell 2003].

Zur Darstellung von vektorbasierten Grafikformaten muss die Anzeige der jeweiligen Grafik auf dem Rechner, auf dem sie dargestellt werden soll, berechnet werden [Kunze 2006]. Diese Tatsache kann zwar bei komplexen Grafiken zu Verzögerungen führen, bringt aber auch den Vorteil mit sich, dass Vektorgrafiken bei verteilten Anwendungen leicht client-seitig verändert werden können [Gulbransen, Bartlett, Bingham, Kachur, Rawlings & Watt 2002]. Somit ist die **Animation** von vektorbasierten Grafiken deutlich leichter zu realisieren als bei pixelbasierten Formaten. Vor allem ist es möglich, beispielsweise den Pfad einer Animation oder eine Auswahl zu animierender Objekte **interaktiv** festzulegen.

Eine weitere Besonderheit von Vektorgrafikformaten wie SVG oder Flash ist die Möglichkeit zur **Einbettung von Scriptcode** [Kunze 2006]. Über diesen Weg kann beispielsweise Kommunikation mit externen Komponenten realisiert werden. Ausgelöst werden können die eingebetteten Scriptbestandteile durch verschiedene Events wie das Laden einer Datei oder Nutzeraktionen. Ein Vorteil bei der Auslösung von Events durch Nutzeraktionen wie beispielsweise das Anklicken einer Grafik ist, dass die jeweilige Aktion Objekten in der Grafik und nicht nur Regionen auf einem Bild zugeordnet werden kann.

#### 7.4.1.1.3 SVG vs. Flash

Flash und SVG sind derzeit die am weitesten verbreiteten Vektorgrafikformate zur Darstellung multimedialer Inhalte im WWW [Technical Advisory Service for Images 2006]. Die Formate sind sich in vieler Hinsicht ähnlich; beide ermöglichen Hyperlinking, unterstützen Animationen, sind zoomfähig, bieten Datenbankkonnektivität und sind scriptingfähig [House & Pearlman 2003]. Die Formate weisen jedoch auch genügend Unterschiede auf, um nicht in direkter Konkurrenz miteinander zu stehen [Teague & Campbell 2003]. Der aktuelle Abschnitt vergleicht die beiden Formate und arbeitet dabei heraus, welches sich besser zur Realisierung des im Rahmen der vorliegenden Arbeit entwickelten Navigationskonzeptes eignet.

Einer der deutlichsten Unterschiede zwischen Flash und SVG ist die **Verbreitung von Wiedergabesoftware**. Flash ist ein proprietäres Format das zunächst von der Firma Macro-

media entwickelt wurde und mit der Übernahme von Macromedia durch Adobe an Adobe übergegangen ist [Kunze 2006]. Die Wiedergabe von Flash ist mit dem ebenfalls proprietären Adobe Flash Player möglich. Dieser wurde in der Anfangsphase der Verbreitung von Flash von vielen Browserherstellern direkt mit dem Webbrowser ausgeliefert. Mittlerweile wird das Format auf so vielen Webseiten eingesetzt, dass sich Nutzer das Plugin in der Regel selbst installieren [Kunze 2006]. In der einen oder anderen Version ist Flash heute auf schätzungsweise 96 Prozent aller internettauglichen Rechner in den USA zu finden, SVG nur auf 11,3 Prozent [27]. Nur wenige Webbrowser wie der Mozilla Firefox oder Opera unterstützen in neueren Versionen SVG, allerdings nicht im vollen Umfang der 1.1 Spezifikation [Kunze 2006]. Darüber hinaus existiert eine Reihe verschiedener Plugins und Stand-Alone-Viewer, wie in [Watt 2001] dargestellt. Die Verbreitung der Plugins hält sich jedoch in Grenzen.
Die Tatsache, dass sehr viele SVG Viewer existieren führt zu einer Reihe von Problemen, da kein Viewer den kompletten SVG 1.1 Standard wiedergeben kann und von Viewer zu Viewer **Unterschiede bei der Abdeckung** der jeweils unterstützen Teilbereiche **des Standards** existieren. Das derzeit umfassendste Plugin ist der Adobe SVG Viewer [Kunze 2006].[75] Neben dem SVG 1.1 Standard existiert allerdings noch der SVG Tiny Standard, der von einer Reihe von Viewern unterstützt wird. Eine Übersicht über die SVG-Unterstützung in Web-Browsern ist in [8] zu finden. Da es sich bei Flash im Gegensatz zu SVG nicht um einen offenen Standard handelt, wird eine neue Flash Version immer nur im Zusammenhang mit dem zugehörigen Player entwickelt. Bei Flash treten derartige Komplikationen daher nicht auf.
Ein weiterer Unterscheid zwischen Flash und SVG besteht bezüglich der **Interoperabilität mit anderen Formaten**. SVG kann mit einer Reihe existierender Standards und Technologien wie HTML, JavaScript und CSS kombiniert werden. Eine Besonderheit des Formats ist dabei, dass durch den selben Script-Code gleichzeitig Elemente in HTML und in SVG manipuliert werden können [Teague & Campbell 2003]. Im Fall der vorliegenden Arbeit ist es dadurch sogar möglich, eine in HTML eingebettete Instanz des Real Players aus SVG fernzusteuern (siehe 7.4.1.4.3). Interaktion mit Flash basiert im Gegensatz dazu auf der proprietären Sprache ActionScript und bietet nur minimale Unterstützung für andere Sprachen und Standards [Teague & Campbell 2003].
Flash erlaubt jedoch die **Einbettung multimedialer Inhalte**. Eine Auflistung der unterstützten Audio- und Video-Formate ist in [Held, Ullrich, Neumann & Winter 2004] zu finden. In der Version 1.1 sieht die Spezifikation des SVG-Standards dagegen keine Einbettung multimedialer Inhalte vor [Teague & Campbell 2003]. Mit dem Adobe SVG Viewer ist es allerdings über eine Namespaceerweiterung möglich, Audiodateien abzuspielen [Teague & Campbell 2003].
Einen Nachteil von SVG stellt die mangelnde **Streamingfähigkeit** des Formates dar. Ein SVG-Dokument kann daher so lange nicht angezeigt werden, bis das gesamte Dokument geladen ist. Das Flash Format ist im Gegensatz dazu streamingfähig [Held, Ullrich, Neumann & Winter 2004]. Wie in [Kunze 2006] gezeigt, ist es jedoch in SVG möglich, Inhalte sukzessive nachzuladen oder bereits vor ihrer Verwendung zu übertragen (Precaching). Durch derartige Maßnahmen lässt sich die mangelnde Streamingfähigkeit des Formates weitestgehend ausgleichen.
Auch die Struktur der beiden Formate SVG und Flash unterscheidet sich grundlegend. **Flash ist ein Binärformat, SVG hingegen XML-basiert** [Held, Ullrich, Neumann & Winter

[75] Adobe spielte eine maßgebliche Rolle bei der Entwicklung und Verbreitung von SVG [Teague & Campbell 2003]. In wie weit Adobe diese Rolle nach der Übernahme von Macromedia und damit dem Erwerb des Flash Formates weiter spielt, ist jedoch fraglich. So hat Adobe beispielsweise angekündigt, den Support für den Adobe SVG Viewer am 1.1.2008 einzustellen [29].

2004]. Der Aufbau des SVG-Formates als XML bringt dabei eine Reihe von Vorteilen mit sich. So sind Elemente eines SVG-Dokumentes in Textform repräsentiert und können damit prinzipiell durch Suchmaschinen indiziert werden [Probets, Mong, Evans & Brailsford 2001]. Auch bezüglich der Verarbeitung und Erzeugung von SVG erweist sich die XML-basierte Struktur des Formates als hilfreich [Probets, Mong, Evans & Brailsford 2001], da mittlerweile eine Vielzahl von Werkzeugen zur Erzeugung und zum Umgang mit XML existiert. So kann SVG beispielsweise während der Darstellung mittels Scriptsprachen manipuliert werden (vgl. 7.4.1.3). Hinzu kommt die Tatsache, dass SVG als XML-basiertes Textformat von Menschen gelesen werden kann [Held, Ullrich, Neumann & Winter 2004].
Ein weiterer Vorteil des Textformates von SVG ist die Tatsache, dass SVG von beliebigen Screen Readern verarbeitet werden kann und damit im Gegensatz zu Flash auch **für Blinde eine Möglichkeit zur Nutzung** von SVG besteht [Teague & Campbell 2003]. Bei Vorlesungsaufzeichnungen fällt dieser Faktor allerdings nicht ins Gewicht, da Folien hier ohnehin von einem synchronisierten Audiostrom begleitet werden.
In Abwägung der in diesem Abschnitt beschriebenen Faktoren ist im Rahmen der vorliegenden Arbeit **SVG zur Darstellung** der folienbasierten Inhalte von Vorlesungs-aufzeichnungen **gewählt worden**. Gründe dafür sind die XML-basierte Struktur des Formates und die Möglichkeit zur einfachen Interoperabilität mit anderen Sprachen und Formaten. SVG vereinfacht damit die Entwicklung des im Rahmen dieser Arbeit vorgeschlagenen Benutzerinterfaces. Bezüglich des Einsatzes im Regelbetrieb hat sich die Verwendung von SVG jedoch als durchaus problematisch erwiesen.

#### 7.4.1.2 Real Video

Zur Wiedergabe des Vorlesungsvideos wird, wie bereits zuvor erwähnt, das Format Real Video eingesetzt. Die Begründung hierfür ist in erster Linie historischer Natur. An der Universität Osnabrück wurde das Format bereits vor Beginn dieser Arbeit zur Aufzeichnung von Vorlesungen eingesetzt und die entsprechende Software zum Betrieb von Real Streaming Servern sowie Lizenzen für die nötige Real Producer Software standen in ausreichender Zahl zur Verfügung.
Ein weiterer Grund für die Verwendung von Real Video ist die Vielzahl von Möglichkeiten zur Synchronisation von Real Video mit anderen Medien. Insgesamt bieten sich drei verschiedene Wege. So kann der Real Player in HTML-Seiten eingebettet werden und über JavaScript ferngesteuert werden [19]. Eine Alternative dazu stellt die Synchronisation von Real mit wechselnden in sich statischen Inhalten wie Webseiten oder auch nicht animierten SVG-Dokumenten über Real Events dar [18]. Darüber hinaus ist der Real Player dazu in der Lage, SMIL wiederzugeben [19]. Eine kurze Betrachtung der Eignung dieser drei Synchro-nisationsmöglichkeiten für eine Implementation der in Kapitel 1 dargestellten Form wird in 7.4.1.4 gegeben.

#### 7.4.1.3 JavaScript

Die Sprache JavaScript ist zur Darstellung interaktiver Inhalte in Webbrowsern konzipiert. Standardisiert ist die Sprache unter dem weniger häufig verwendeten Namen ECMAScript [ECMA 1999]. JavaScript wird von allen wichtigen Webbrowsern unterstütz, wenn auch in leicht unterschiedlichen Ausprägungen. Die Sprache bietet die Möglichkeit HTML- und XML-Dokumente über deren Document Object Model (DOM) zu manipulieren [7].
Das DOM ist eine vom W3C definierte Programmierschnittstelle, die auch im Rahmen der SVG-Spezifikation vorgesehen ist. Die scriptbasierte Manipulation von SVG über dessen DOM wird allerdings nicht in allen Webbrowsern unterstützt. In [8] ist eine Übersicht darüber zu finden, in welchen Webbrowsern SVG ggf. unter Verwendung eines Plugins durch JavaScript manipuliert werden kann. Wie bereits in 7.4.1.2 erwähnt, kann JavaScript

darüber hinaus zu Steuerung des Real Players verwendet werden. Auch hier werden allerdings wiederum nicht alle Browser und Betriebsysteme unterstützt.

```
...
// the constructor of the class
function StepSlideCtrl()
{

      this.maxTime = 10*60*60*1000; // 10h

      this.steps = new Array(); // steps are the main units on the
                                // time line
      this.slides = new Array(); // all slides including
                               // intermediate steps
      this.init = stepSlideCtrl_init; // assign function

      this.setMaxTime = function(t) // assign function
            {this.maxTime = t;}

...
}

// the function initiates internal arrays slides() and steps().
// It uses the the array stepSequenceList.
function stepSlideCtrl_init()
{
       // the main problem: slides can be flipped through
       // on the time line in reverse order
       // i.e. after step 15 the next step can be 14 and not
       // necessarily 16. Rnadom jumps are also possible.
...
}
...
```

**Code-Beispiel 7-4: Codebeispiel zur Verwendung von Objektmethoden in JavaScript**

JavaScript bietet eine Reihe objektorientierter Eigenschaften [Kühl 2002]. So können in JavaScript Exceptions verarbeitet und Objektmethoden definiert werden. Die Definition von Objektmethoden gestaltet sich allerdings etwas unintuitiv. Code-Beispiel 7-4 zeigt an den mit den Worten *assign function* kommentierten Stellen zwei Möglichkeiten, Methoden einer Klasse von Objekten zuzuordnen. Bei der Arbeit mit dem Code bedeutet dies, dass Methodenköpfe nicht durch einfache Volltextsuche gefunden werden können. Die Suche nach dem Methodennamen unter Auslassung der Methodenklammern führt jedoch in jedem Fall zu einem Treffer. Eine gut kommentierte Beispielsammlung zur objektorientierten Programmierung mit JavaScript ist in [38] zu finden.
JavaScript ist schwach typisiert [Sanders 2001]. Damit ist implizite Typumwandlung möglich und eine Typüberprüfung findet erst zur Laufzeit statt. Diese Eigenschaften erleichtern zwar die Programmierung, erschweren aber unter Umständen die Fehlersuche. Darüber hinaus unterstützt JavaScript kein Multithreading [Flanagan 1997] und erlaubt damit keine parallele Abarbeitung von nebenläufigen Programmteilen. Zwar kann in den meisten Browsern durch Ausführung von JavaScript in unterschiedlichen Browserfenstern die parallele Ausführung von JavaScript-Code erreicht werden. In diesem Fall arbeiten jedoch unterschiedliche Programmzähler auf unterschiedlichen Ausführungsstacks und es existiert keine Möglichkeit, die Ergebnisse der Aufrufe zusammenzuführen. Die Nebenläufigkeit existiert daher auf Ebene des Sprachinterpreters des Browsers, nicht auf Ebene der Sprache JavaScript. Es handelt sich dabei also nicht um MultiThreading in JavaScript.

Dieser Nachteil kann teilweise durch die Methoden `setInterval()` und `setTimeout()` ausgeglichen werden. Die Methode `setTimeout()` erlaubt den zeitverzögerten Aufruf einer Methode und die Methode `setInterval()` bewirkt den wiederholten Aufruf einer Methode in festgelegten Zeitabständen. Ist der Ausführungsstack durch einen anderen Methodenaufruf, wie beispielsweise das Laden eines größeren Dokumentes blockiert, wird der durch `setTimeout()` oder `setInterval()` geplante Aufruf erst nach Abarbeitung des aktuellen Aufrufs ausgeführt.

#### 7.4.1.4 Synchronisation von Film und Folien

Wie bereits in 7.4.1.2 erwähnt, bietet Real Video drei verschiedene Wege, um ein Real Video mit statischen und teilweise sogar dynamischen Dokumenten zu synchronisieren. Im Folgenden werden kurz Erfahrungen diskutiert, die im Rahmen der vorliegenden Arbeit mit den verschiedenen Synchronisationsmethoden gemacht wurden und die letztendlich zur Verwendung von Javascript zur Synchronisation geführt haben.

##### 7.4.1.4.1 Real Events

Als Zusatz zum Real Producer und zum Helix Producer wird von der Firma Real das Hilfsprogramm `rmevents.exe` ausgeliefert. Mit diesem Programm können Real Video Dateien und Real Audio Dateien mit zusätzlichen Instruktionen versehen werden, die die Anzeige eines HTML-Dokumentes oder eines beliebigen anderen im Browser darstellbaren Medienformates zu festgelegten Zeitpunkten beim Abspielen des Videos bewirken [18]. Bei der Erstellung der so genannten *Eventdatei* kann für einzelne Events festgelegt werden, wo das jeweilige Dokument angezeigt werden soll, beispielsweise in einem neuen Browserfenster oder der sogenannten *Media Browser Pane* des Real Players. Bei Verwendung des Real Players als eingebettete Komponente in einem Frame eines HTML-Dokuments kann das Ziel eines Events auch in einem anderen Frame des Dokumentes angezeigt werden. Dazu muss bei der Definition des Events der Name des Zielframes angegeben werden.
Im Rahmen der vorliegenden Arbeit wurde die Verwendung von Real Events zur Synchronisation von Video und SVG-Folien in der Anfangsphase des Projektes eingesetzt. Dabei wurden Erfahrungen mit zwei verschiedenen Varianten gesammelt. Zur Durchführung des in 3.2 beschriebenen Szenarios wurde die Synchronisation des Videos mit SVG-Folien in einem externen Browserfenster eingesetzt. Dabei wurden die Folien und das Video jeweils auf einem separaten Grafikausgang wiedergegeben. In diesem Fall funktionierte die Synchronisation von Folien und Video problemlos. In der zweiten Variante wurden Real Events in einer frühen Version des in Kapitel 1 beschriebenen Webinterfaces zur Synchronisation einer in ein HTML-Dokument eingebetteten Instanz des Real Players mit SVG-Folien in einem benachbarten Frame verwendet. Dabei fielen wiederholt Zeitverzögerungen von mehreren Sekunden auf. Da diese Zeitverzögerungen nicht deterministisch auftreten und in ihrer Zeitdauer variieren, besteht keine Möglichkeit sie durch eine Anpassung der Eventzeiten zu beseitigen. Zur Implementation des in Kapitel 1 beschriebenen Webinterfaces ist eine Synchronisation mittels Real Events daher nicht geeignet.

##### 7.4.1.4.2 SMIL

Das Akronym SMIL steht für *Synchronized Multimedia Integration Language*. SMIL ist eine XML-basierte Sprache zur Synchronisation, Anordnung und Steuerung von multimedialen Inhalten wie Audio- und Videodateien, Text und Grafik. Zur Wiedergabe von SMIL existiert eine Reihe von Playern, eine Übersicht ist in [43] zu finden.
Da im Rahmen der vorliegenden Arbeit das Real Video Format zu Wiedergabe der Videoaufzeichnung verwendet wird, wurde die Synchronisation von Video und Folien für das virtPresenter Web Interface nur im Real Player getestet. Zur Anzeige von Inhalten, die nicht

direkt im Real Player wiedergegeben werden können, verwendet der Player den Internet Explorer. Diese Tatsache ist dem Nutzer gegenüber insoweit transparent, dass die Ausgabe des Internet Explorers im Real Player selbst angezeigt wird. Durch Ausgabe des Browsertyps mittels JavaScript ist jedoch erkennbar, dass die jeweiligen Inhalte nicht im Real Player sondern in einer gekapselten Instanz des Internet Explorers wiedergegeben werden. Da aus dieser gekapselten Instanz heraus keine Möglichkeit zum scriptgesteuerten Zugriff auf die umgebende Instanz des Real Players existiert, können interaktive Elemente in einer Anzeigefläche eines SMIL-Dokumentes nicht scriptgesteuert mit anderen Elementen des selben Dokumentes kommunizieren. Zwar ist ein Umweg der Kommunikation über einen Server möglich, doch schließt sich durch ein solches Vorgehen die Verwendung des Interfaces von einem Offline-Medium aus. Auch ist ein derartiger Kommunikationsweg vergleichsweise langsam und beeinträchtigt beispielsweise im Falle der Anzeige von Markierungslinien struktureller Elemente auf der Zeitleiste (vgl. 6.2.2) merklich die Reaktivität der Oberfläche.
Hinzu kommt die Tatsache, dass die Anzeige von SVG im Real Player mit dem Adobe SVG Viewer bei einigen Versionen des Real Players nicht funktioniert. In Versionen des Real Players, in denen die Anzeige von SVG möglich ist, können allerdings sogar Animationen im SVG mit dem Video synchronisiert werden. Da die Anzeige von SVG im Real Player jedoch nicht zuverlässig funktioniert und darüber hinaus keine scriptbasierte Kommunikation zwischen einzelnen Dokumenten, die in ein SMIL-Dokument eingebettet sind, möglich ist, ist SMIL zur Realisierung des in Kapitel 1 beschriebenen Interfaces nicht geeignet.

##### 7.4.1.4.3 Synchronisation per Javascript

Die Synchronisation von Ereignissen wie Folienwechseln mit einem Real Video, das in einer in ein HTML-Dokument eingebetteten Instanz des Real Players abgespielt wird, ist auf zwei verschiedene Weisen möglich. Zum einen bietet der Real Player die Möglichkeit, Events an JavaScript-Objekte zu senden. Der zugrunde liegende Mechanismus basiert dabei auf Callback-Methoden [17]. Die Palette der Ereignisse, zu denen der Real Player Events versendet, umfasst eine Reihe von Zustandsänderungen. So werden Ereignisse wie Änderungen des Abspielstatus (Stop, Play, Pause usw.) oder auch Positionsveränderungen beim Abspielen einer Datei übermittelt. In einer Testversion des Interfaces haben sich Events, die bei der Änderung des Spielzustandes versendet werden, allerdings als nicht zuverlässig erwiesen, was vermutlich mit dem zur Anzeige der SVG-Folien verbundenen Rechenaufwand zusammen hängt. Auch die Verwendung von Ereignissen, die bei Änderungen der Abspielposition versendet werden, hat sich als nicht zweckmäßig erwiesen, da der Real Player diese Ereignisse in einer zu hohen Taktrate versendet und damit die Ausführung des zugehörigen JavaScript-Codes zu häufig triggert. Dadurch bleibt keine Rechenzeit zur Abarbeitung von Nutzeraktionen mehr und das Interface hängt sich auf.
Die zweite Möglichkeit, Ereignisse per JavaScript mit einem Real Video, das in einer eingebetteten Instanz des Real Player abgespielt wird, zu synchronisieren besteht in einer aktiven Abfrage durch Polling des jeweiligen Zustandes des Real Players. Durch die Methoden `setInterval()` und `setTimeout()` (vgl. 7.4.1.3) kann dabei festgelegt werden, in welchen Zeitabständen die Abfragen stattfinden. Hierdurch kann die mit den Abfragen verbundene Rechenzeit auf ein adäquates Maß reduziert werden.

#### 7.4.1.5 JavaServer Pages zur Implementation serverseitiger Komponenten

Die Nutzerverwaltung des in Kapitel 1 beschriebenen Webinterfaces ist serverseitig realisiert. Diese Designentscheidung ist vor allem durch die Tatsache bedingt, dass zur Umsetzung der sozialen Navigation (vgl. 6.4) die Zugriffsdaten aller Nutzer zentral gehalten werden müssen. Darüber hinaus können durch eine serverseitige Verwaltung von Nutzerdaten Navigationshilfen wie Footprints und Bookmarks für einen Nutzeraccount von

verschiedenen Rechnern aus genutzt werden. Diese ist beispielsweise dann von Vorteil, wenn Studierende die Aufzeichnung in universitätseigenen Rechnerräumen und vom heimischen PC aus nutzen.

Die Implementation einer serverseitigen Nutzerverwaltung ist mit verschiedenen Technologien wie PHP, Servlets oder JavaServer Pages realisierbar. Im Rahmen der vorliegenden Arbeit wurden JavaServer Pages (JSP) gewählt, da diese Technologie mit Java auf eine vollwertige und hoch entwickelte Programmiersprache mit einer Vielzahl von Code-Bibliotheken zurückgreifen kann. Hinzu kommt die Tatsache, dass eingebundener Java-Code in Syntax und Funktionalität eine hohe Ähnlichkeit zu C# aufweist, wodurch sich die Wartbarkeit des Projektes als Ganzes erhöht. Eine eingehendere Betrachtung der Vorteile von JSP ist in [Hall 2002] zu finden.

JSP sind eine Weiterentwicklung von Servlets. Servlets generieren wie auch PHP serverseitig HTML-Code, der an den Client gesendet und dort angezeigt wird. Der Unterschied zwischen Servlets und JSP besteht darin, dass der HTML-Code der generierten Webseiten nicht durch Print-Anweisungen aus dem Java-Code heraus erzeugt werden muss. Ein JSP-Dokument besteht aus HTML, in das durch bestimmte Tags gekennzeichneter Java-Code eingebunden werden kann.[76] Zur Trennung von Form und Inhalt kann der verwendete Java-Code auch in Form von Beans ausgelagert werden.

Auf einem Webserver können verschiedene JSP-Applikationen ausgeführt werden, die jeweils über eine eigene URL zugänglich sind. Wird eine JSP-Seite aufgerufen, ohne dass für die Verbindung zwischen Browser und JSP-Applikation bereits eine Session existiert, wird im Regelfall automatisch eine neue Session angelegt. Dieses Verhalten kann jedoch auch deaktiviert werden [Chan 2002]. Eine der Hauptaufgaben von Web-Applikation besteht darin, Daten verschiedener Nutzer voneinander getrennt zu verwalten [Hall 2002]. In einer JSP-Applikation werden Daten daher meist sessionspezifisch verwendet, es besteht aber auch die Möglichkeit, Daten zwischen Sessions auszutauschen. In einigen Fällen kann es ressourcenbedingt auch sinnvoll sein, Daten und Objekte über die Grenzen von Webapplikationen hinweg zu verwenden. Zu diesem Zweck müssen Klassen, die das Interface `javax.Servlet.ServletContext` implementieren, die Methode `getContext(String url)` zur Verfügung stellen. Da die applikationsübergreifende Verwendung von Ressourcen jedoch in einigen Fällen ein Sicherheitsrisiko darstellen kann, ist die Methode in einigen Webservern wie beispielsweise dem Apache Tomcat standardmäßig deaktiviert und gibt nur den Wert `null` zurück. Im Tomcat kann die Verwendung der Methode `getContext(String url)` durch das Setzen der Variable `crossContext` in der Serverkonfiguration auf `true` erreicht werden. Alternative, jedoch weniger mächtige Methoden zum reinen Datenaustausch zwischen Applikationen sind Cookies sowie die Verwendung von Datenbanken oder des Dateisystems des Webservers.

### 7.4.2 Details der Implementation

Die Implementation der Wiedergabekomponente gliedert sich grob in die zwei Bereiche clientseitige und serverseitige Implementation. Clientseitig ist das Nutzerinterface implementiert, das die synchronisierte Wiedergabe von Video und Folien ermöglicht und die in Kapitel 1 beschriebenen Interaktionsmöglichkeiten realisiert. Die serverseitige Implementation umfasst die Verwaltung von Nutzerdaten einzelner Nutzer sowie die Aufbereitung von Zugriffsdaten zur sozialen Navigation. Im weiteren Verlauf dieses Abschnittes wird zunächst die clientseitige Implementation und anschließend die Implementation der serverseitigen Komponenten des Wiedergabeinterfaces vorgestellt.

[76] Dabei handelt es sich jedoch nicht um einen funktionellen Unterschied, da JSP vom Webserver in Servlets umgewandelt werden und damit lediglich eine Programmiererleichterung darstellen.

#### 7.4.2.1 Clientseitige Implementation

Das in Kapitel 1 beschriebene Interface ist unter Verwendung von HTML und JavaScript realisiert. Zur Unterteilung des Interfaces in die einzelnen Hauptkomponenten wie Folienüberblick, Anzeigefläche der Hauptfolie, Kontrollleiste und Anzeigefläche des Videos werden dabei Frames genutzt. Interaktive Bedienelemente wie die interaktive Zeitleiste, Bedienknöpfe des Interfaces oder auch maussensitive Bestandteile animierter SVG-Folien im Folienüberblick sind an JavaScript-Methoden gekoppelt. Die gesamte Programmlogik des Interfaces ist dabei zentral in einer Reihe von JavaScript-Klassen zusammengefasst, die in die jeweiligen HTML-Dateien eingebunden werden. Zur Ansteuerung des Real Players wird dessen JavaScript Schnittstelle verwendet (vgl. 7.4.1.2). Änderungen in den SVG-Dokumenten werden mittels DOM-Manipulation der jeweiligen Dokumente über JavaScript realisiert. Die Synchronisation von Video und SVG ist wie bereits in 7.4.1.4.3 beschrieben mittels JavaScript implementiert.
In den folgenden beiden Unterabschnitten werden zunächst Details zur Verwendung SVG-basierter Bestandteile des Interfaces diskutiert. Anschließend wird die clientseitige Implementation zu Verwaltung von Nutzungsdaten beschrieben.

##### 7.4.2.1.1 SVG-basierte Komponenten

Neben den Folien im Folienüberblick und der Hauptfolie ist auch die Zeitleiste ein SVG-Dokument. Dadurch kann die dynamische Darstellung von Footprints und die Einblendung von Zusatzinformationen wie Folien- und Animationsgrenzen über Änderungen des DOM des SVG-Dokumentes erreicht werden. Vor allem bei der Darstellung von Footprints erweist sich dieses Vorgehen als deutlich weniger rechenintensiv als beispielsweise die Darstellung und Aktualisierung einer pixelbasierten Zeitleiste über JavaScript. Als weiterer Vorteil kommt die Tatsache hinzu, dass in SVG die Darstellung von halbtransparenten Objekten möglich ist. Damit kann die Farbtonabstimmung von Footprints zur Darstellung mehrfach angezeigter Passagen der Aufzeichnung durch eine überlagernde Darstellung der zugehörigen Footprints sehr einfach realisiert werden.
Die Verwendung von SVG bietet jedoch nicht nur Vorteile. Als ein entscheidender Nachteil ist die in 7.4.1.1.3 geschilderte Problematik der unzureichenden Abdeckung des SVG-Standards durch viele Viewer zu nennen. Da die entwickelte Applikation eine Reihe von Eigenschaften des SVG-Standards ausnutzt, die nur im Adobe SVG Viewer (ASV) nutzbar sind, ist das Interface auch nur im Internet Explorer mit installiertem ASV lauffähig. Zur Detektion und Meldung fehlender oder nicht unterstützter Plugins wird daher bei der Initialisierung des Interfaces ein entsprechender Codeblock ausgeführt.
Auch der ASV bringt eine entscheidende Schwäche mit sich, die jedoch erst bei intensiverer Nutzung auffällt. Komplexere SVG-Dokumente führen beim ASV zu vergleichsweise langen Renderingzeiten im Bereich von mehreren hundert Millisekunden. Beim Laden von einzelnen SVG-Dokumenten auf Webseiten fällt dieser Zeitdauer kaum in Betracht. Bei der Aktualisierung der Hauptfolie im Folienfenster des Interfaces fällt diese Verzögerung jedoch unangenehm auf. Problematisch werden längere Wartezeiten bei der Initialisierung des Folienüberblicks. Da hier je nach Gestaltung der verwendeten PowerPoint-Präsentation in der Regel ca. zehn bis vierzig Titelfolien, in Einzelfällen aber auch mehr, dargestellt werden müssen, können sich die Renderingzeiten zu Werten bis in den Minutenbereich aufaddieren. Beim Laden des Interfaces sind derartige Wartezeiten jedoch nicht zumutbar. Bei der Initialisierung des Interfaces werden daher zunächst Platzhalter angezeigt, die nach und nach durch die Titelfolien ersetzt werden. Das Interface ist dabei bereits vor dem Laden der Titelfolien nutzbar.
Ein weiteres Problem ergibt sich in diesem Zusammenhang aus der dynamischen Visualisierung struktureller Kontexte (vgl. 6.1.2). Da hierzu SVG-Dokumente in der Folienüber-

sicht ausgetauscht werden, treten auch hier immer wieder kurze Wartezeiten auf. Beim Überfahren von Navigationshilfen wie Bookmarks oder Pfeilen zur strukturbasierten Navigation („<<, >>“) sind diese Wartezeiten kaum störend. Probleme ergeben sich allerdings bei der Navigation mit Brushing in der Zeitleiste. Hier sollte die Visualisierung struktureller Grenzen zeitgleich mit der Vorschau von Folien im Folienüberblick stattfinden. Außerdem sollten jeweils die zeitlichen Grenzen der Folie und Folienanimation angezeigt werden, die zur aktuellen Mauszeigerposition gehören. Wird nun beim Überspringen einer Folien- oder Animationsgrenze zunächst der Folienüberblick aktualisiert, ist die Kennzeichnung der Folien- und Animationsgrenzen mehrere hundert Millisekunden lang fehlerhaft. Durch eine Umkehrung der Abarbeitungsreihenfolge der Anweisungen zur Aktualisierung von Folienüberblick und Markierungen auf der Zeitleiste lässt sich dieses Problem zwar für einen Aktualisierungsschritt umgehen, bei schnellem Brushing tritt das Problem jedoch beim Wechsel zum nächsten Animationsschritt, bzw. zur nächsten Folie auf.
Eine einfache Lösung durch Entkopplung der Aktualisierungsschritte von Zeitleiste und Folienüberblick ist aufgrund des Fehlens von Multithreading in JavaScript (vgl. 7.4.1.3) nicht möglich. Als Alternative ist daher eine Methode implementiert, die die Aktualisierung des Folienüberblicks verwaltet und mit Hilfe der Methode `setTimeout()` verzögert. Aktualisierungsschritte werden dabei nur dann ausführt, wenn sie nicht bereits durch nachfolgende Aktualisierungsschritte ungültig geworden sind. Bei schnellem Brushing wird durch diese Methode zwar eine korrekte Aktualisierung des Folienüberblicks verhindert, allerdings bleibt die Anzeige der strukturellen Grenzen auf der Zeitleiste aktuell. Auch eine grobe Orientierung auf dem Folienüberblick bleibt bestehen, da hier die Titelanzeige der betreffenden Folie farblich hervorgehoben wird und dieser Vorgang keiner Zeitverzögerung unterliegt. Bei langsamem Brushing wird die Aktualisierung des Folienüberblicks nicht behindert, da die verwendeten Wartezeiten im Centisekundenbereich liegen.

#### 7.4.2.1.2 Verwaltung von Nutzungsdaten

In einem herkömmlichen Webbrowser werden Nutzungsdaten, die zur Verwaltung von Footprints und Bookmarks nötig sind vom Browser selbst verwaltet. Aufgrund der zeitbasierten Beschaffenheit von Vorlesungsaufzeichnungen kann bei der Implementation des im Rahmen dieser Arbeit entwickelten Interfaces allerdings nicht auf die entsprechenden Funktionalitäten des verwendeten Browsers zurückgegriffen werden. Hinzu kommt die Tatsache, dass Nutzungsdaten zur sozialen Navigation ohnehin serverseitig verwaltet werden. Die zur Realisierung von Footprints und Bookmarks nötigen Nutzungsdaten werden daher in der Interface-Applikation selbst gesammelt und verwaltet.
Die Verwaltung der entsprechenden Nutzerdaten ist dabei an die Elemente zur Darstellung von Footprints (die interaktive Zeitleiste) und Bookmarks gekoppelt. Immer dann, wenn beispielsweise durch einen Navigationssprung, durch Pausieren oder durch das Beenden der Webseite (Unload-Event) das Zeichnen eines Footprints beendet wird, wird diese Änderung an den Server übermittelt und dort gespeichert.
Zusätzlich existiert eine Variante des Interfaces, die zum lokalen Betrieb von CD oder DVD aus konzipiert ist. In dieser Variante ist keine soziale Navigation möglich und die Nutzungsdaten werden als Cookies in dem Browser abgelegt, der zur Wiedergabe der Aufzeichnung verwendet wird.

### 7.4.2.2 Serverseitige Implementation

Zur Bereitstellung des Real Videos kann existierende Standardsoftware wie der Helix DNA Server verwendet werden. Auch die SVG-Folien können von einem herkömmlichen Webserver aus angeboten werden. Zur serverseitigen Verwaltung von Nutzungsdaten des Interfaces ist jedoch die Entwicklung von eigener Software nötig. Zu diesem Zweck sind

einige Komponenten des Interface Codes in JSP geschrieben. Zur Aufbereitung und Verwaltung der Daten zur sozialen Navigation wird darüber hinaus das Softwaretool ANT und eine MySQL Datenbank genutzt. Als Webserver wird der Apache Tomcat verwendet.
Im Folgenden wird zunächst ein kurzer Überblick über wesentliche Punkte bei der Realisierung der JSP-basierten Komponenten der serverseitigen Implementation gegeben. Anschließend wird die Aufbereitung von individuellen Nutzungsdaten zur sozialen Navigation dargestellt.

#### 7.4.2.2.1 JSP-basierte Komponenten

Die Implementation der verwendeten JSP Komponenten gestaltet sich vergleichsweise geradlinig. Durch das Interface übermittelte Änderungen von Nutzungsdaten werden im Dateisystem gespeichert und beim erneuten Laden des Interfaces aus dem Dateisystem gelesen und an den Client übermittelt. Die Daten zur sozialen Navigation werden aus einer Datenbank gelesen. Die Aufbereitung von einzelnen Nutzungsdaten zur sozialen Navigation wird in einem ausgelagerten Prozess mittels ANT realisiert (vgl. 7.4.2.2.2).
Um jedem Nutzer den Zugriff auf seine eigenen Daten und nur auf diese Daten zu ermöglichen,[77] ist eine einfache Nutzerverwaltung unter Verwendung einer Datenbank implementiert. Da Datenbanken oft die Anzahl eingehender Verbindungen beschränken, werden Datenbankverbindungen für alle JSP-Applikationen unterschiedlicher Aufzeichnungen auf einem Server zentral in einer zusätzlichen JSP-Applikation verwaltet (vgl. 7.4.1.5 zur applikationsübergreifenden Verwendung von Ressourcen in JSP).

#### 7.4.2.2.2 Aufbereitung von Nutzungsdaten zur sozialen Navigation

Wie in 6.4 dargestellt, können in der Komponente zur sozialen Navigation die Nutzungsdaten aller Nutzer akkumuliert über alle Wochen des Semesters oder auch zu spezifischen Einzelwochen angezeigt werden. Die angezeigten Nutzungsdaten werden dazu wie bereits in 7.4.2.2.1 erwähnt aus einer Datenbank gelesen, wohingegen die Nutzungsdaten einzelner Nutzer aufgrund der Größe und Struktur der Datenmengen im Dateisystem des Servers verwaltet werden. Zur Aufbereitung der Nutzungsdaten wird in regelmäßigen Zeitabständen, die über serverseitige Einstellungen festgelegt werden können, ein ANT Script angestoßen. Das Script summiert die Nutzungsdaten jeder einzelnen Aufzeichnung auf und legt diese in der Datenbank ab. Da bei der Nutzung von Aufzeichnungen über mehrere Semester hinweg die einzelnen Semesterwochen mehrmals durchlaufen werden, wird bei der Berechnung der wochenweisen Nutzungsdaten das Anfangsdatum des jeweiligen Semesters berücksichtig. Zu diesem Zweck ist in der Datenbank eine Tabelle mit Semesteranfangsdaten vorhanden. Die Tabelle ist nicht automatisch generiert und muss daher in regelmäßigen Abständen, beispielsweise einmal im Jahr, nachgepflegt werden. Davon abgesehen ist die Aufbereitung von Nutzungsdaten zur sozialen Navigation vollständig automatisiert.

## 7.5 Zusammenfassung und Einordnung in den weiteren Verlauf

In diesem Kapitel wurde das Zusammenspiel der Komponenten des implementierten Systems zur Aufzeichnung und Wiedergabe von Vorlesungsmitschnitten dargestellt. Darüber hinaus wurde der Aufbau der Komponenten geschildert. Im nachfolgenden Kapitel werden Einzelheiten bei der Erzeugung und Aufbereitung der Vorlesungsfolien im Wiedergabeformat SVG beschrieben

[77] Im Rahmen der sozialen Navigation kann darüber hinaus zwar auch auf die Nutzungsdaten anderer Nutzer zugegriffen werden, jedoch nur in akkumulierter und anonymisierter Form.

.

# 8 Erzeugung und Aufbereitung der Vortragsfolien in SVG

Wie in 7.3 beschrieben, wird in der Nachbearbeitungsphase der Aufzeichnung die PowerPoint-Präsentation in einen Satz von Folien im SVG Format umgewandelt. Um die Durchsuchbarkeit der Folien im Interface und die Verlinkung animierter Folienbestandteile in die Aufzeichnung zu ermöglichen, sind darüber hinaus noch Nachbearbeitungsschritte nötig. Zur Bestimmung der Ansprungpunkte von animierten Folienbestandteilen wird dabei die Protokolldatei der Folienwechsel- und Animationszeitpunkte verwendet. Als letzter Nachbearbeitungsschritt werden die Folien gezippt, um die Größe der resultierenden Dateien zu verringern. Abb. 8-1 gibt einen schematischen Überblick über die Anordnung der Schritte in der Nachbearbeitungsphase.

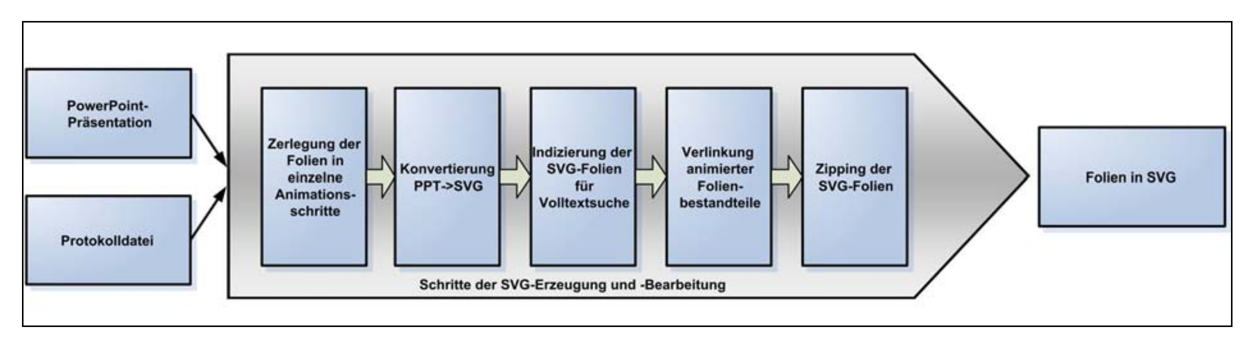



**Abb. 8-1: Erzeugung der SVG-Folien aus PowerPoint und Nachbearbeitungsschritte**

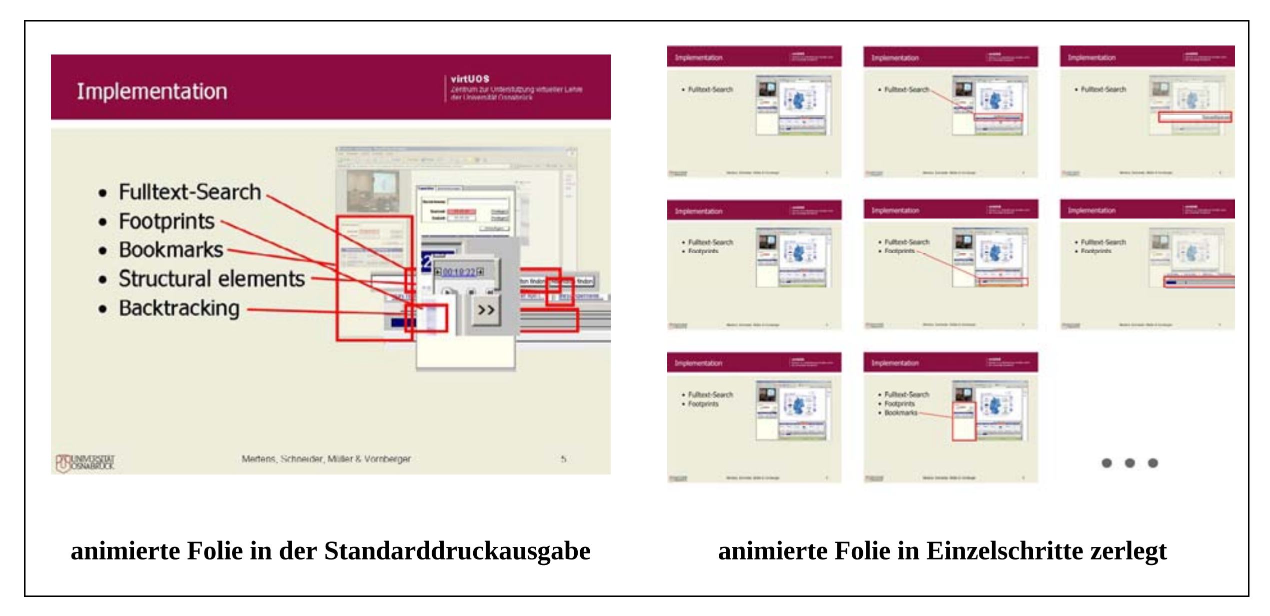
Im vorliegenden Kapitel werden die einzelnen Schritte der Erzeugung und Aufbereitung der Vortragsfolien, von der PowerPoint-Präsentation bis zu den Dokumenten, die letztendlich im Interface verwendet werden, dargestellt.

## 8.1 Zerlegung der PowerPoint-Präsentation in Einzelschritte

Zur eigentlichen Konvertierung von PowerPoint nach SVG wird wie bereits in 7.3.1.3 erwähnt, die Software SVGmaker verwendet. Diese Software nutzt wiederum die Druckausgabe von PowerPoint. Fehler oder Unstimmigkeiten in der Druckausgabe von PowerPoint werden daher direkt an den SVGmaker weitergeleitet. Ein sehr auffälliges Problem ist, dass Animationsschritte in einer Folienpräsentation von PowerPoint beim Drucken nicht berücksichtigt werden. Stattdessen druckt PowerPoint Folien lediglich in einer Art Editionsansicht, in der alle auf der Folie vorhandenen Objekte gleichzeitig dargestellt werden. Abb. 8-2 verdeutlicht den Unterschied zwischen einer Folie in mehreren Animationsschritten und der Editionsansicht der Folie.

Die Verwendung von Folien, die mit Hilfe der Druckausgabe erzeugt wurden, ist daher im Fall von animierten PowerPoint-Präsentationen wegen einer Reihe von Gründen auszuschließen. So sind beispielsweise Objekte, die während der Präsentation aus der Folie ausgeblendet werden, in der Druckansicht zu sehen. Damit können Objekte, die einen Teil der Folie absichtlich zu einem frühen Teil der Präsentation verdecken, nicht entfernt werden. Außerdem kann hier ein Wirrwarr von Objekten auf der Folie entstehen, die in der Präsentation in einzelnen Animationsschritten auf der Folie auftauchen und wieder verschwinden. Einzeln und zu unterschiedlichen Zeitpunkten dargestellt können diese Objekte durchaus Sinn machen, in ihrer Masse betrachtet sind sie jedoch bestenfalls verwirrend, wie auch aus Abb. 8-2 ersichtlich. Außerdem geht die Reihenfolge von Animationsschritten, die Hinweise über die Ablaufreihenfolge eines dargestellten Sachverhaltes geben, verloren. Der schrittweise Aufbau einer Folie ist dadurch nicht mehr zu erkennen. Das Fehlen dieser zeitlichen Information kann das Verständnis der Darstellung erheblich beeinträchtigen. Hinzu kommt

die Tatsache, dass durch den Wegfall der Darstellung von Animationsschritten die gezeigte Folie in vielen Fällen nicht mehr dem Redetext des Dozenten entspricht. Dadurch, dass der Dozent von erscheinenden und verschwindenden Objekten redet, während auf dem Bildschirm nur eine starre, nicht animierte Folie zu sehen ist, geht damit die Konsistenz von Wort und Bild verloren. Aus diesen Gründen ist es für die im Rahmen der vorliegenden Arbeit entwickelte Software essentiell, dass Animationsschritte bei der Wiedergabe der Aufzeichnung dargestellt werden können.




**animierte Folie in der Standarddruckausgabe** **animierte Folie in Einzelschritte zerlegt**

**Abb. 8-2: Vergleich der Standarddruckausgabe von PowerPoint und einer Einzelansicht von Folienanimationsschritten**

Um mit dem SVGmaker eine animationsschrittgetreue Umwandlung von PowerPoint-Präsentationen erzeugen zu können, ist daher ein Zerlegungsprogramm entwickelt worden. Das Programm verarbeitet eine PowerPoint-Präsentation so, dass jeder Animationsschritt als separate Folie über die Druckausgabe ausgegeben werden kann. Im weiteren Verlauf dieses Abschnittes wird zunächst die grundlegende Arbeitsweise dieses Programms dargestellt, anschließend wird eine Reihe von Detailproblemen mitsamt der zu ihrer Lösung verwendeten Ansätze vorgestellt. In diesem Abschnitt werden die implementierten Komponenten auf algorithmischer Ebene betrachtet. Zur Ausführung der Algorithmen wird, wie in Kapitel 7 beschrieben, die von PowerPoint bereitgestellte COM-Schnittstelle unter C# genutzt. Eine Dokumentation des verwendeten PowerPoint-Objektkataloges ist unter [13] zu finden. Für eine Diskussion technischer Problemlösungen im Zusammenhang mit der Fernsteuerung von PowerPoint sei auf das vorangehende Kapitel verwiesen.

### 8.1.1 Arbeitsweise des Zerlegungsalgorithmus

Im Zerlegungsalgorithmus wird zu jedem Animationsschritt eine PowerPoint-Folie erzeugt, die optisch genau der Ansicht entspricht, die in der Präsentation nach Ausführung des Animationsschrittes angezeigt wird. Zu diesem Zweck wird zunächst wie in 8.1.1.1 dargestellt der Folienhintergrund auf einer Folie erzeugt. Anschließend werden auf diesem Hintergrund alle animierten Folienobjekte in der zum jeweiligen Animationsschritt aktuellen Ansicht arrangiert. Dieser Teil der Implementation wird in 8.1.1.2 beschrieben.

#### 8.1.1.1 Erzeugung der Folienhintergründe

Prinzipiell existieren zwei Wege, Folienhintergründe zu erzeugen, auf denen die graphischen Objekte einer animierten Folie zur Druckausgabe angeordnet werden können. Diese Wege

sind zum einen das Erzeugen von Kopien der Ursprungsfolie sowie zum anderen die Nutzung des Folienhintergrundes und der Folienposition der Ursprungsfolie, wie im weiteren Verlauf dieses Abschnittes erläutert. Im Folgenden wird zunächst diskutiert, welche Gründe gegen die Verwendung von Kopien der Ursprungsfolie sprechen. Anschließend wird der in der Implementation verwendete Ansatz zur Nutzung des Folienhintergrundes der Ursprungsfolie vorgestellt.

#### 8.1.1.1.1 Kopien der Ursprungsfolie

Eine Kopie der Ursprungsfolie kann per automatisiertem Copy und Paste entweder in einer zweiten Präsentation oder in der Präsentation, in der auch die ursprüngliche Folie enthalten ist, erzeugt werden.
Die Verwendung einer zweiten Präsentation ist dabei allerdings nicht ratsam, da Formatierungsinformationen wie Schriftgröße, Schriftfarben und Layout, die in der ursprünglichen Präsentation enthalten sind, verloren gehen. Diese Attribute können zwar programmgesteuert angepasst werden, aufgrund der Vielzahl von Parametern und der Tatsache, dass diese Parameter nicht zentral dokumentiert sind, ist die korrekte Abdeckung aller Attribute in einem Programm jedoch nur sehr schwer realisierbar. Abb. 8-3 zeigt exemplarisch einige Veränderungen, die beim Kopieren von Folienelementen aus einer Präsentation in eine Präsentation mit anderen Voreinstellungen auftreten können. Besonders auffällig sind in diesem Beispiel die Änderung der Formatierung des Titels, die Farbänderungen bei drei der vier dargestellten Schlüssel und die Positionsänderung einiger Textbausteine.

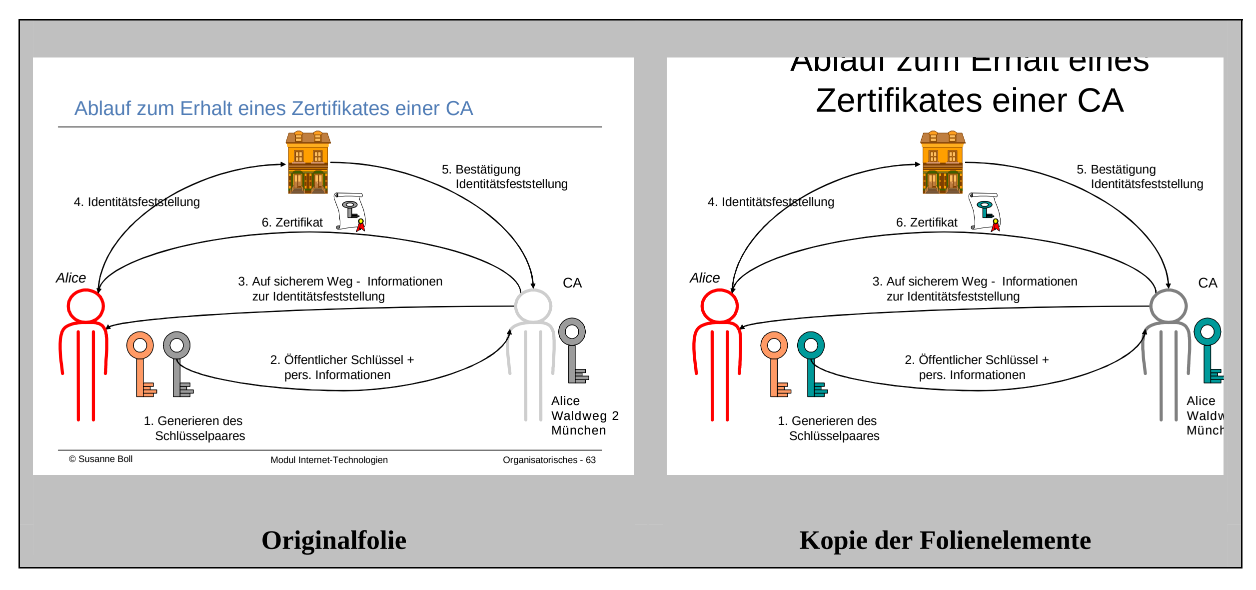



**Abb. 8-3: Veränderung der Darstellung von Folienelementen bei Wechsel der Präsentation**

Auch die Anfertigung von Kopien innerhalb der Ursprungspräsentation geht mit einer Reihe von Komplikationen einher. Vor allem Felder, die durch den Folienmaster beeinflusst werden und Felder, die automatisch abhängig von der Position der Folie innerhalb der Präsentation aktualisiert werden, stellen hier oft Problemfälle dar. Derartige Felder müssen zunächst im Folienmaster gelöscht und anschließend in Position und Erscheinungsbild genau nachgebildet werden. Auch hier stellt die Vielzahl der zu beachtenden Parameter und deren fehlende Dokumentation das Hauptproblem dar.

#### 8.1.1.1.2 Nutzung der Position der Ursprungsfolie in der Ursprungspräsentation

Alternativ zu einer Kopie der Ursprungsfolie in einer anderen Präsentation oder an einer anderen Folienposition in der Ursprungspräsentation kann zum Ausdruck von Animationsschritten auch die Position der Ursprungsfolie genutzt werden. Auch in diesem Fall müssen

die Objekte, die auf der Folie im jeweiligen Animationsschritt angezeigt werden, zunächst an einem anderen Ort zwischengespeichert werden. In der Implementation werden zum Zwischenspeichern zusätzliche Folien in der Ursprungspräsentation genutzt, wie in Abb. 8-4 gezeigt.
Dabei wird die animierte Folie zunächst expandiert, d.h. für jeden einzelnen Animationsschritt wird an der letzten Folienposition der Präsentation eine Hilfsfolie erzeugt, in der der Zustand der Folie zu dem betreffenden Animationsschritt festgehalten wird. Die hierzu verwendeten Algorithmen werden in 8.1.1.2.2 beschrieben und sind so gestaltet, dass Einstellungen zur Foliendarstellung für die Hilfsfolie übernommen werden. Nach der Expansion der Folie wird für jeden Animationsschritt die Hilfsfolie mit der Ursprungsfolie vertauscht und die Hilfsfolie wird über den SVGmaker gedruckt. Dadurch, dass die Hilfsfolie sich zum Druckzeitpunkt an der Position der Ursprungsfolie befindet, werden alle Felder automatisch mit den richtigen Werten aktualisiert und müssen nicht nachträglich manipuliert werden. Nach Abarbeitung aller Animationsschritte der Folie werden die durch die Expansion der Folie entstandenen Hilfsfolien gelöscht, da PowerPoint die auf den Hilfsfolien enthaltenen Objekte kopiert und dadurch bei größeren Foliensätzen sehr schnell Speicherplatzprobleme entstehen können. Anschließend wird der beschriebene Vorgang für die nachfolgende Folie wiederholt, bis alle Einzelschritte der Präsentation gedruckt sind.

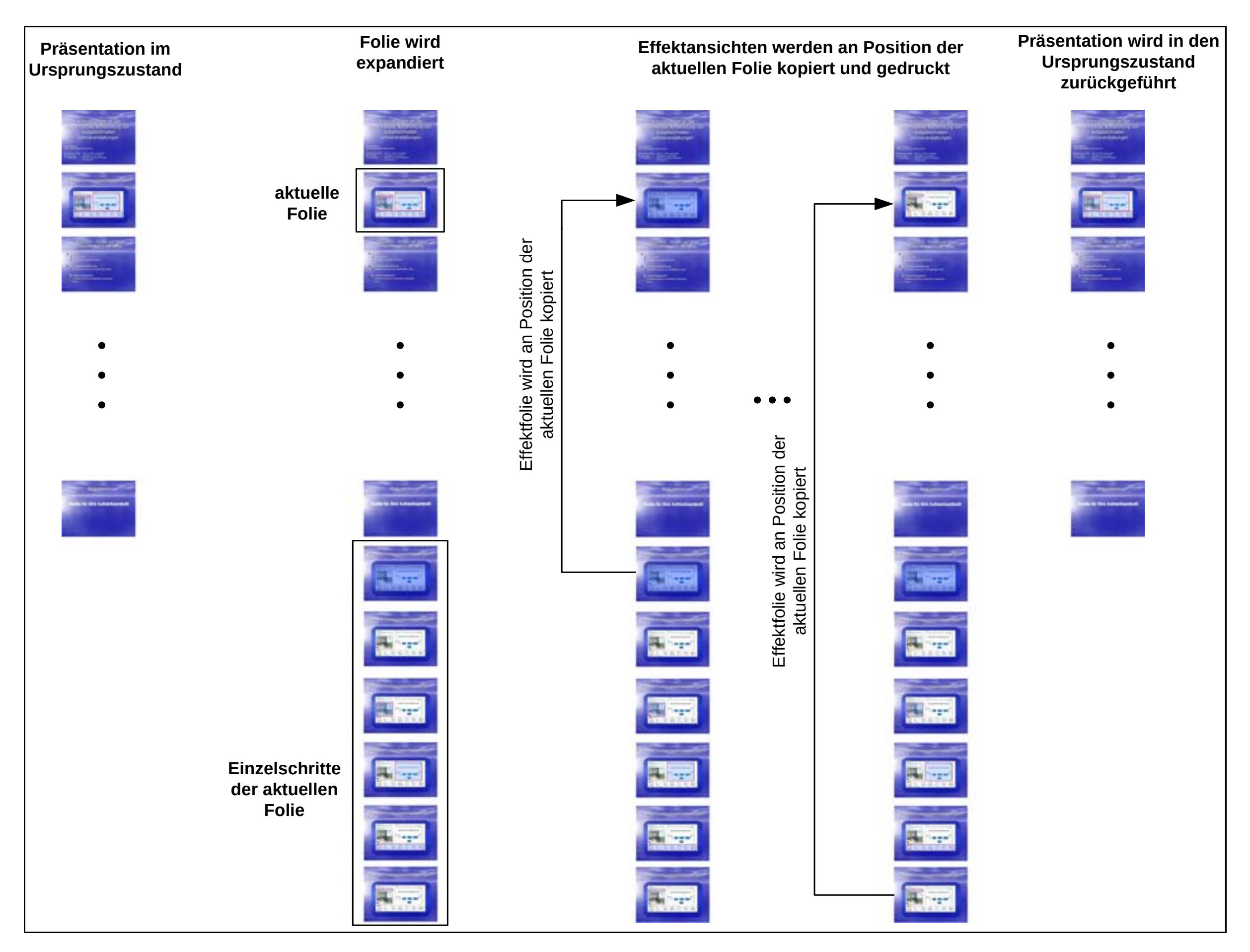



**Abb. 8-4: Einzelschritte bei der Nutzung der Ursprungsfolie zum Ausdruck von Animationsschritten**

### 8.1.1.2 Sequenzierung animierter Folien

Um eine Folie so in Einzelschritte zu zerlegen, dass jeder Animationsschritt auf einer separaten Hilfsfolie abgebildet werden kann, wird zunächst der Zustand aller Objekte auf der Anfangsansicht der Folie in der Bildschirmpräsentation bestimmt. Danach werden aufbauend

auf der Anfangsansicht sukzessive die Einzelschrittfolien für alle Animationsschritte der Folie erzeugt. In den folgenden Unterabschnitten wird zuerst die implementierte Vorgehensweise zur Bestimmung des Anfangszustandes einer Folie erläutert. Zusätzlich werden im ersten Unterabschnitt Vorverarbeitungsschritte zur Verarbeitung komplexerer Animationskategorien dargestellt. Im zweiten Unterabschnitt wird die Erzeugung der jeweiligen Einzelschrittfolien behandelt. Der dritte Unterabschnitt ist der Diskussion der verwendeten Algorithmen und der Darstellung verschiedener Details der Foliensequenzierung gewidmet.

#### 8.1.1.2.1 Bestimmung des Anfangszustandes einer Folie und Vorverarbeitung komplexer Effekte

Zur Bestimmung des Anfangszustandes von animierten Folien wird die Ablaufbeschreibung der Animationseffekte der Folie geparst. Diese Ablaufbeschreibung ist in dem Folienobjekt vom Typ `Microsoft.Office.Interop.PowerPoint.Slide` in der Variablen `TimeLine.MainSequence` gespeichert.
Bei der Abbildung von Effekten auf Einzelschrittfolien sind die vier Grundkategorien *Eingang*, *Hervorgehoben*, *Beenden* und *Animationspfade* zu unterscheiden. In jeder dieser Kategorien bietet PowerPoint eine Reihe von Effekten, die teilweise spezielle Anforderungen an die Zerlegung stellen. Effekte der Kategorien Eingang und Beenden sind dabei vergleichsweise leicht nachzubilden, da sie durch das Hinzufügen bzw. Entfernen der animierten Objekte simuliert werden können. Effekte der Kategorien Hervorgehoben und Animationspfade sind hingegen aufwändiger nachzubilden. Nach einer Reihe von Vorverarbeitungsschritten werden jedoch auch Effekte dieser Kategorien auf das Erscheinen und Entfernen von Objekten abgebildet.
Das Erscheinen und Verschwinden von Objekten wird beim Parsen der Ablaufbeschreibung der Animationseffekte in zwei Hashtabellen festgehalten, die anschließend zur Konstruktion der Einzelschrittfolien verwendet werden. In den Hashtabellen werden jeweils die Objekte verwaltet, die auf der Folie zu sehen sind (Hashtable `InSlide`) bzw. nicht auf der Folie zu sehen sind (Hashtable `OutSlide`). Initial wird zunächst für alle Objekte angenommen, dass sie vor dem ersten Animationsschritt auf der Folie zu sehen sind. Während des Parsens der Ablaufbeschreibung der Animationsreihenfolge wird diese Annahme ggf. korrigiert. Da bei der Abarbeitung von Animationseffekten der Kategorien Eingang und Beenden teilweise Animationseffekte aus der `TimeLine.MainSequence` gelöscht werden, die darin enthaltenen Informationen jedoch in nachgelagerten Verarbeitungsschritten benötigt werden, wird eine Kopie dieses Objektes genutzt.
In den folgenden Unterabschnitten werden die zur Simulation der vier Kategorien implementierten Lösungsansätze für jede Kategorie einzeln vorgestellt. Dabei werden zunächst die Vorverarbeitungsschritte zur Behandlung von Effekten der Kategorien Animationspfade und Hervorgehoben beschrieben. Anschließend werden die Vorgehensweisen zur Behandlung von Effekten der Kategorien Eingang und Beenden dargestellt, die auch in der Abarbeitungsreihenfolge der Implementation zeitlich nachgelagert sind.

##### 8.1.1.2.1.1 Effektkategorie Animationspfade

Bei diesem Effekttyp verschiebt sich das betroffene Objekt entlang eines vordefinierten Animationspfades. Die betroffenen Eigenschaften sind daher die x- und y-Koordinaten des Objektes. Die anzuwendenden Veränderungen sind in dem zugehörigen Objekt vom Typ `Microsoft.Office.Interop.PowerPoint.Effect` (im weiteren Verlauf mit Effect-Objekt bezeichnet) gespeichert. Da die Positionswerte von Shape-Objekten in PowerPoint (`Microsoft.Office.Interop.PowerPoint.Shape`) und die Angaben, die in den betreffenden Effect-Objekten gespeichert werden, nicht in der selben Maßeinheit vorliegen, wird ein heuristisch ermittelter Korrekturfaktor angewendet.

Zur Erzeugung der Momentaufnahmen einzelner Folienschritte bei Pfadanimationen wird der Effekt jeweils mit zwei verschiedenen Objekten dargestellt, die den Zustand des betreffenden Objektes vor bzw. nach dem Animationseffekt darstellen. Der Effekt selbst wird dabei durch zwei direkt nacheinander ausgelöste Effekte ersetzt. Der erste dieser Effekte lässt das Objekt, das den Zustand des Objektes vor dem Animationsschritt darstellt verschwinden. Der direkt danach ausgeführte zweite Effekt lässt das zweite Objekt, das den Zustand des Objektes nach dem Animationsschritt darstellt, erscheinen.
Da beim Kopieren eines Shape-Objektes alle zugehörigen Effekte auf die Kopie übertragen werden, müssen alle durch das Kopieren entstandenen Effekte gelöscht werden. Aufgrund der Tatsache, dass das ursprüngliche Shape-Objekt nach dem Kopieren durch zwei verschiedene Objekte dargestellt wird, müssen alle Effekte, die sich auf das Ursprungsobjekt beziehen und in der `TimeLine.MainSequence` nach dem aktuell behandelten Pfadeffekt angeordnet sind, auf das neu erzeugte Objekt übertragen werden.
Durch dieses Vorgehen ergeben sich dann Komplikationen, wenn im weiteren Verlauf eine weitere Pfadanimation auf das Shape-Objekt angewendet wird. In diesem Fall beziehen sich deren Koordinaten auf die ursprüngliche Position des Ursprungsobjektes. Die Zuordnung von Kopie und Original, also dem Shape-Objekt, das die Position nach der Animation darstellt und dem Shape-Objekt, das die Position vor der jeweiligen Pfad-Animation darstellt, wird daher in einer Hashtabelle gespeichert. Falls ein Shape-Objekt mehrmals mit Pfadanimationen bewegt wird, wird das jeweilige Ursprungsobjekt durch einen rekursiven Algorithmus ermittelt, der die Kette von Ursprungsbeziehungen bis zum jeweils ersten Ursprungsobjekt durchläuft.

#### 8.1.1.2.1.2 Effektkategorie Hervorgehoben

Im Gegensatz zu den im vorigen Abschnitt behandelten Pfadeffekten, haben nicht alle Effekte aus dieser Kategorie nachhaltige Auswirkungen auf die Eigenschaften des betroffenen Shape-Objektes. Beispiele für Hervorhebungseffekte, die Objekteigenschaften permanent verändern, sind die Effekttypen Größenveränderungen und Transparenzänderungen. In der Implementation sind derzeit nur Größenveränderungen realisiert, andere Effekttypen lassen sich leicht in den Programmcode einfügen, wurden jedoch bisher nicht von Nutzern des Systems verwendet. Andere Effekte wie Wellenanimationen oder Flimmern haben keine nachhaltigen Auswirkungen auf die Darstellungseigenschaften des Objektes.
Hervorhebungseffekte, die keine bleibenden Auswirkungen auf Objekteigenschaften haben, können bezüglich der Foliendarstellung ignoriert werden. Da sie jedoch während der Aufzeichnung von der Aufzeichnungskomponente (vgl. 7.2.2.2) genau so gezählt werden wie andere Effekte, können sie bei der Folienzerlegung nicht einfach gelöscht werden. Um diese Effekte dennoch darstellungsneutral verarbeiten zu können, werden sie in einem Vorverarbeitungsschritt durch einen Dummy-Skalierungseffekt ersetzt, der das entsprechende Objekt um den Faktor 1 skaliert, also in der Größe nicht verändert.
Hervorhebungseffekte, die Objekteigenschaften permanent verändern, werden wie Effekte der Kategorie *Animationspfade* durch zwei direkt nacheinander folgende Effekte ersetzt, die das betreffende Objekte durch eine Kopie ersetzen, bei der die jeweiligen Eigenschaften entsprechend angepasst sind. Da Pfadanimationen implementationsbedingt zeitlich vor Hervorhebungseffekten bearbeitet werden, sind an dieser Stelle auch Kopien des Objektes zu berücksichtigen. Durch einen während der Verarbeitung der entsprechenden Pfadanimation erfolgten Eintrag in eine Hashtabelle können diese Objekte lokalisiert werden. Bei mehreren Pfadanimationen auf demselben Objekt werden zur Propagierung der Objektveränderung rekursiv alle zugehörigen Objekte behandelt.

#### 8.1.1.2.1.3 Effektkategorie Eingang

Objekte, deren erste Animation durch einen Animationseffekt der Kategorie *Eingang* beschrieben wird, werden bis zur Ausführung dieses Effektes nicht auf der Folie angezeigt. Diese Information ist nicht explizit in den Animationseinstellungen des betreffenden Objektes gespeichert sondern nur implizit in der Reihenfolge der Animationseffekte in der `TimeLine.MainSequence` des Folienobjektes. Wird beim Parsen der Reihenfolge der Animationseffekte daher ein Einganseffekt für ein Objekt gefunden, wird das Objekt von der Hashtabelle `InSlide` in die Hashtabelle `OutSlide` verschoben. Um Verwechslungen bei der Abarbeitung nachfolgender Animationseffekte der Kategorie *Beenden* auf dem selben Objekt vorzubeugen (vgl. 8.1.1.2.1.4), werden anschließend alle Animationseffekte auf diesem Objekt aus der verwendeten Kopie der `TimeLine.MainSequence` gelöscht.

###### 8.1.1.2.1.4 Effektkategorie Beenden

Objekte, deren erster Animationseffekt in der `TimeLine.MainSequence` ein Effekt der Kategorie *Beenden* ist, werden bei Ausführung dieses Effektes von der Folie entfernt. Vor Ausführung des Effektes sind sie allerdings auf der Folie zu sehen und müssen daher in der Hashtabelle `InSlide` verbleiben. Damit die Objekte nicht bei der Abarbeitung nachfolgender Effekte der Kategorie Eingang zur Herstellung des Initialzustandes der Folie in die Hashtabelle `OutSlide` verschoben werden, werden analog zur Abarbeitung von Effekten der Kategorie Eingang alle nachfolgenden Objekte in der `TimeLine.MainSequence`, die auf dem betreffenden Objekt arbeiten aus der verwendeten Kopie gelöscht.

##### 8.1.1.2.2 Aktualisierung der Animationsschritte

Nach Herstellen des Initialzustandes werden anhand der `TimeLine.MainSequence` schrittweise Einzelschrittfolien zur Darstellung der einzelnen Animationsschritte erzeugt. Dabei wird mit Hilfe der Hashtabellen `InSlide` und `OutSlide` bestimmt, welche Objekte sich auf der Folie befinden.

Je nachdem, ob es sich bei dem jeweils behandelten Effekt um einen Effekt der Kategorie *Eingang* oder *Beenden* handelt, wird das animierte Objekt von `OutSlide` nach `InSlide` verschoben oder umgekehrt. Im dem zwar wenig sinnvollen aber theoretisch möglichen Fall, dass zwei Effekte der Kategorie *Eingang* oder *Beenden* hintereinander auf das selbe Objekt angewendet werden, bleiben die Einträge der Hashtabellen unverändert. Bei der Abarbeitung des nachfolgenden Animationsschrittes wird der erzeugte Zustand der Hashtabellen weiter genutzt. Pro Eintrag in der `TimeLine.MainSequence` der Folie wird also nur ein Objekt von einer der beiden Hashtabellen in die jeweils andere verschoben. Einen Sonderfall bei der Erstellung der Einzelschrittfolien stellen Animationseffekte dar, die gleichzeitig ausgeführt werden. In diesem Fall wird die Methode zum Update der Hashtabellen rekursiv für alle gleichzeitigen Animationseffekte aufgerufen.

Nach dem Update der Hashtabellen wird die Originalfolie ans Ende der Folienpräsentation kopiert. Auf der Kopie der Folie werden alle Objekte gelöscht, die nicht in `InSlide` enthalten sind. Die Objekte, die auf der Folie verbleiben werden ggf. so manipuliert, dass sie in der Darstellung dem aktuellen Animationsschritt entsprechen (siehe 8.1.1.2.1.1, 8.1.1.2.1.2 und 8.1.1.2.3.1).

In PowerPoint wird die Tiefenordung (ZOrder) der dargestellten Objekte defaultmäßig durch die Reihenfolge bestimmt, in sie in eine Folie eingefügt werden. Das zuletzt eingefügte Objekt wird dabei vor allen anderen platziert.

Die Reihenfolge, in der Objekte in eine PowerPointfolie eingefügt worden sind, spiegelt sich auch in der Shapes-Auflistung der Folie wider. Da bei der Zerlegung der Bewegungsschritte einer Folie in Einzelfolien jedoch animierte Objekte nicht aus der Shapes-Auflistung, sondern aus der Effektreihenfolge (`TimeLine.MainSequence`) herauskopiert werden, kann es in den Zielfolien zu einer Umordnung der Tiefenreihenfolge kommen. Eine Umordnung der

Tiefenordnung der verschiedenen Objekte kann zu einem nicht intendierten Überlappungsverhalten führen wie in Abb. 8-5 dargestellt.

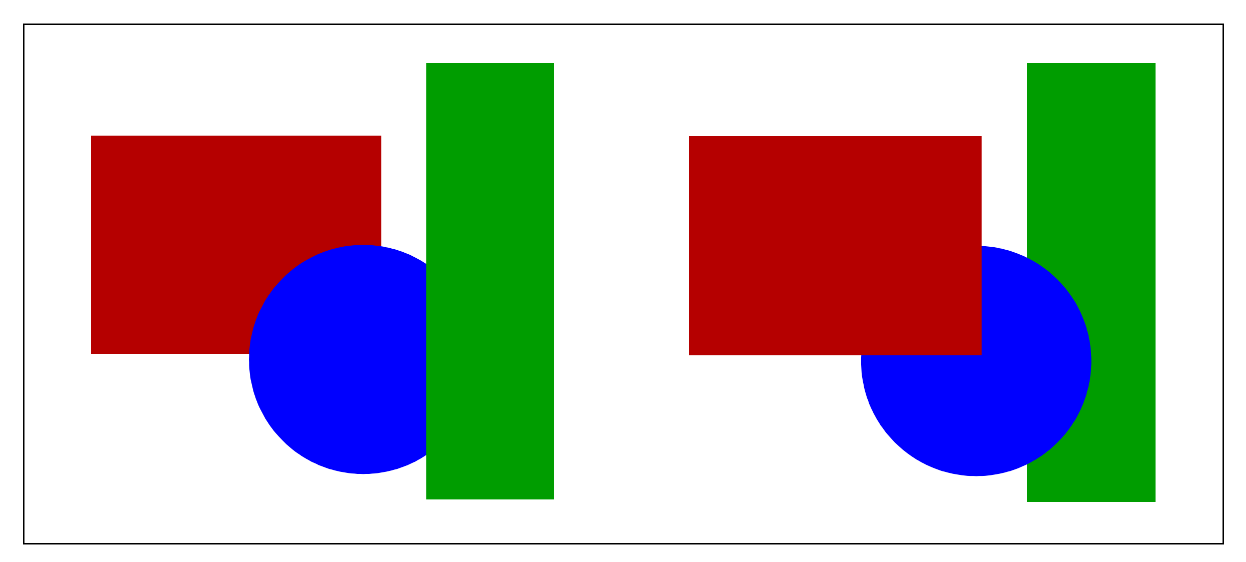


**Abb. 8-5: Vergleich von zwei Objektgruppen mit unterschiedlicher Tiefenanordnung der Objekte**

Um diesem Problem entgegen zu wirken, müssten die Objekte in der Zielfolie nach der Tiefenordnung der Ursprungsfolie sortiert werden. Ein derartiges Vorgehen ist jedoch nur durch automatisiertes Copy und Paste der Einzelobjekte zu realisieren. Copy und Paste ist bei der Fernsteuerung von PowerPoint über COM allerdings sehr rechenintensiv und führt leicht zu instabilem Verhalten. Aus diesem Grund wird die Folie als komplettes Objekt kopiert und nicht benötige Bestandteile werden aus der Kopie gelöscht.

#### 8.1.1.2.3 Details der Implementation

Bei der im vorigen Abschnitt beschriebenen Aktualisierung der Animationsschritte ist eine Reihe von Sonderfällen zu berücksichtigen. So können Effekte mit so genannten *Nacheffekten* versehen werden, die die Darstellung des animierten Objektes nach Ausführung des Effektes verändern. Bei der Animation von Textfeldern ist es möglich, nur einzelne Zeilen zu animieren und einige Effekte lassen sich aufgrund fehlender Plugins auf dem Server nicht konvertieren. Weitere Probleme stellen Effektinformationen dar, die nicht auf der betreffenden Folie, sondern nur im Folienmaster gespeichert sind und so nur schlecht konkreten Objekten zugeordnet werden können. Im weiteren Verlauf des aktuellen Abschnittes wird eine relevante Auswahl derartiger Sonderfälle zusammen mit den implementierten Lösungsansätzen beschrieben.

##### 8.1.1.2.3.1 Nacheffekte

PowerPoint bietet die Möglichkeit, direkt nach einem Effekt einen *Nacheffekt* anzustoßen. Dieser *Nacheffekt* kann nur das vom Effekt betroffene Objekt verändern. Welcher Art dieser Nacheffekt ist, wird durch den Wert `Effect.Shape.AnimationSettings.AfterEffect` im Effect-Objekt festgelegt. In PowerPoint gibt es fünf verschiedene Nacheffekte:

- `msoAnimAfterEffectDim` verändert die Farbe des Objekts.
- `msoAnimAfterEffectHide` lässt das Objekt nach dem Auftauchen wieder verschwinden.
- `msoAnimAfterEffectHideOnNextClick` lässt das Objekt beim nächsten Mausklick, synchron mit dem nächsten Effekt, wieder verschwinden
- `msoAnimAfterEffectMixed` ist in PowerPoint XP noch nicht implementiert.
- `msoAnimAfterEffectNone` stellt keinen Nacheffekt dar.

Diese *Nacheffekte* sind im Zerlegungsalgorithmus mit Ausnahme des Effekts `msoAnimAfterEffectMixed` implementiert. Bei diesem Effekttyp handelt es sich jedoch nur um einen Platzhalter, der dazu dient, Rückwärtskompatibilität für Dateien nachfolgender PowerPoint-Versionen zu erreichen. Sinnvoll kann die Behandlung dieses Effekttyps also erst mit Kenntnis nachfolgender PowerPoint-Versionen implementiert werden. Um auch *Nacheffekte* auf Gruppierungen von Objekten abzudecken, sind die Methoden zur Abarbeitung von *Nacheffekten* rekursiv implementiert.

#### 8.1.1.2.3.2 Objekte mit Textinhalt

Objekte mit Textinhalt werden oft nicht als ein einzelnes Objekt, sondern absatz-, wort- oder buchstabenweise animiert. Es ist daher möglich, dass ein Objekt mehrmals in der Ablaufreihenfolge der Animationseffekte einer Folie vorkommt. In den Variablen `Shape.AnimationSettings.TextLevelEffect` und `Shape.AnimationSettings.TextUnitEffect` enthält jedes animierte Objekt Informationen darüber, ob und wenn ja, wie der Inhaltstext des Objektes animiert wird. Da wort- und buchstabenweise Animationen zwar schrittweise gezeigt werden, alle Schritte innerhalb eines Absatzes allerdings innerhalb der Ablaufdauer eines Effektes stattfinden, kann die Behandlung dieser Effektkategorien in die Behandlung absatzweiser Animationen integriert werden.
Um die Anzeige von einzelnen Absätzen in den in 8.1.1.2.2 beschriebenen Algorithmus zu integrieren, werden die Shape-Objekte der jeweils verwendeten Animationseffekte nicht direkt in den Hashtabellen `InSlide` und `OutSlide` gespeichert, sondern in Wrapper-Objekten. Diese Wrapper-Objekte können zusätzlich Informationen über die angezeigten Absätze aufnehmen.
Zur Darstellung der nach einem Animationsschritt angezeigten Absätze in einem Objekt mit Textinhalt werden in dem Textfeld des Objektes alle nicht angezeigten Absätze gelöscht. Um zu verhindern, dass die nachfolgenden Absätze nach oben verschoben werden, werden entstandene Lücken durch leere Absätze ersetzt, die nur aus einem Zeilenumbruch bestehen. Da die entstandenen Lücken mehre Textzeilen umfassen können, wird die Textgröße der verwendeten Leerabsätze durch ein binäres Suchverfahren so ermittelt, dass Folgeabsätze an der selben Position beginnen wie auf der Originalfolie.
Ein weiteres Problem bei der Löschung von Absätzen aus einem Textfeld kann die automatische Anpassung der Schriftgröße durch PowerPoint darstellen. Dieser Mechanismus passt die Schriftgröße von Text in Textfeldern in einigen Fällen automatisch so an, dass das komplette Feld ausgefüllt wird. Da diese Anpassung nach dem Löschen und Anpassen von Absätzen oft nicht in den Originalschriftgrößen der angezeigten Textbausteine resultiert und eine Korrektur einzelner Textteile eine fehlerhafte PowerPointseitige Anpassung anderer Textteile bewirken kann, wird die Größe aller angezeigten Textbausteine über einen speziellen Korrekturalgorithmus reguliert.
In diesem Algorithmus werden absatzweise die Schriftgrößen aller Absätze mit aufsteigender Absatznummer korrigiert. Da jeweils bei der Korrektur eines Absatzes die Schriftgrößen in anderen Absätzen PowerPoint-seitig verändert werden können, werden nach der Korrektur eines Absatzes alle vorangehenden Absätze erneut überprüft. Damit ergibt sich für den Korrekturalgorithmus eine Laufzeit in der in Formel 8-1 beschriebenen Form, wobei $n$ für die Anzahl der Absätze steht.

$$\sum_{i=1}^{n} i$$

**Formel 8-1: Laufzeit des Korrekturalgorithmus'**

Nach der gaußschen Summenformel ergibt sich damit eine Laufzeitabschätzung von *O(n2)*. Aufgrund dieser hohen Laufzeit des Korrekturalgorithmus' ist eine unnötige Ausführung vermeiden. Der Algorithmus wird daher nur dann ausgeführt, wenn Textbestandteile durch die bereits beschriebene PowerPointseitige Anpassung verfälscht worden sind. Zum Vergleich der Textbestandteile der Originalfolie und der Kopie wird jeweils die Schriftgröße der beiden Objekte verwendet.

#### 8.1.1.2.3.3 Fehlende Informationen zum Effekttyp

In einigen Fällen werden beim Import von Foliensätzen Animationseffekte nicht mit Informationen über ihre Zugehörigkeit zu einem Effekttyp versehen. Die Abfrage der Objektvariablen `EffectType` führt in diesem Fall zu einer unbehandelten Ausnahme. Da der Effekttyp an verschiedenen Stellen des Zerlegungsalgorithmus' benötigt wird, werden derartige Effekte im Vorfeld durch Standardeffekte ausgetauscht. Dabei wird der neu hinzugefügte Effekte mit der Variablenbelegung des auszutauschenden Effektes versehen. Bezieht sich der betroffene Effekt bei einem Textobjekt nur auf einen bestimmten Absatz, animiert PowerPoint beim Austauschen das gesamte Textobjekt, wobei für jeden Absatz des Textobjektes ein zusätzliches Textobjekt entsteht. In diesem Fall werden die zusätzlich entstandenen Effekte gelöscht.

#### 8.1.1.2.3.4 Soundeffekte

Soundeffekte werden naturgemäß nicht auf der Folie dargestellt und haben daher auch kein eigenes Shape-Objekt. Für diese Art von Effekten muss demzufolge auch keine eigene Folie erzeugt werden. Aus diesem Grund werden diese Effekte im Zerlegungsalgorithmus ignoriert. Da die Schrittnummerierung des Zerlegungsalgorithmus mit der des Listeners übereinstimmen muss (vgl. 7.2.2), werden diese Effekte auch im Listener ignoriert.

#### 8.1.1.2.3.5 Fehlende Plugins

Einige Bildformate benötigen zur Darstellung bestimmte Plugins. Sind diese nicht auf dem Konvertierungsserver installiert, können Bilder in entsprechenden Formaten nicht konvertiert werden. Außerdem wird von der COM-Schnittstelle beim Zugriff auf diese Objekte der Wert null zurückgegeben. Um eine Anpassung der oben beschriebenen Algorithmen an vielen Stellen zu vermeiden, werden Effekte, die sich auf Objekte beziehen, zu deren Darstellung die nötigen Plugins fehlen, sind in einem Vorverarbeitungsschritt durch Dummy-Effekte ersetzt. Einfaches Löschen der betroffenen Effekte ist an dieser Stelle keine sinnvolle Alternative, da Informationen über die Effekte während der Aufzeichnung gesammelt werden und in der Protokolldatei enthalten sind.

#### 8.1.1.2.3.6 Animationen auf dem Folienmaster

Objekte in Platzhaltern wie Text- oder Bildfeldern sind in einigen Präsentationen per Default über den Folienmaster animiert. In diesem Fall sind die entsprechenden Effekte nicht in der `TimeLine.MainSequence` der Folie sondern nur auf dem Folienmaster gespeichert. Um die entsprechenden Informationen dennoch im Rahmen des in 8.1.1.2.2 beschriebenen Algorithmus' verarbeiten zu können, werden in einem Vorverarbeitungsschritt alle Folienobjekte, die über diesen Mechanismus animiert sind mit eigenen Animationseffekten versehen. Anschließend werden alle Animationen aus dem Folienmaster gelöscht.
Die programmgesteuerte Ersetzung der Animationsschritte ist dabei nur dadurch möglich, dass Informationen über Animationen, die durch den Folienmaster gesteuert sind, zwar nicht in der `TimeLine.MainSequence` der Folie aber in den einzelnen Objekten gespeichert sind. Durch ein Vertauschen der Effekte auf dem Folienmaster mit speziellen Marker-Effekten kann so jedes einzelne Objekt auf der Folie, sofern es über den Folienmaster animiert ist,

einem bestimmten Effekt auf dem Folienmaster zugeordnet werden. Anschließend kann ein Effekt für dieses Objekt generiert werden und die `TimeLine.MainSequence` eingefügt werden. Zur Positionsbestimmung des jeweiligen Effektes in der `TimeLine.MainSequence` wird die Tatsache ausgenutzt, dass Effekte, die durch den Folienmaster gesteuert werden immer vor allen anderen Animationseffekten auf der Folie ausgeführt werden.

#### 8.1.1.2.3.7 Importprobleme bei eingebetteten Schriften unter PowerPoint 2003

In der Version 2002 von PowerPoint existiert ein Fehler, der das Einbetten von read-only Schriftarten ermöglicht [24]. PowerPoint 2003 öffnet Präsentationen mit eingebetteten read-only Schriften in einem Modus, der das Bearbeiten der Präsentation verhindert.[78] Wird die Präsentation manuell geöffnet, erscheint ein Dialog, in dem der Nutzer auswählen kann, ob die eingebetteten Schriftarten entfernt werden sollen, um das Bearbeiten der Präsentation zu ermöglichen. Bei Zugriffen über die COM-Schnittstelle wird eine PowerPoint-Datei, die derartige Schriften enthält, automatisch in dem beschriebenen änderungsgeschützten Modus geöffnet. Dies hat zur Folge, dass jeder Versuch, Elemente auf einer Folie zu verändern, verhindert wird. Die in diesem Abschnitt beschriebene Manipulation zur Druckvorbereitung ist in diesem Modus nicht möglich. Um diesen Modus zu umgehen, wird die Präsentation in der Aufnahme-Komponente (vgl. 7.2) ohne eingebettete Schriftarten gespeichert. Durch dieses Vorgehen kann die Folienzerlegung sowohl unter PowerPoint 2002 als auch unter PowerPoint 2003 arbeiten. Da diese Schriftarten in der Regel auf jedem Windows System standardmäßig installiert sind, entsteht zwischen den resultierenden PowerPoint-Präsentationen dabei optisch kein Unterschied.

## 8.2 Konvertierung von PowerPoint-Folien nach SVG

Wie bereits in Kapitel 1 dargestellt, wird zur Konvertierung der durch den Zerlegungsalgorithmus erzeugten Einzelschrittfolien die Software SVGmaker verwendet. Da der SVGmaker als Druckertreiber auf dem Konvertierungsrechner installiert ist, kann er bei der Fernsteuerung von PowerPoint über die COM-Schnittstelle als Drucker ausgewählt werden. Anschließend kann die entsprechende Folie in der Präsentation über die Methode `printSlideAs(...)` gedruckt werden. Die eigentliche Konvertierung von PowerPoint nach SVG gestaltet sich damit denkbar einfach.

Beim Einsatz des SVGmakers zur Umwandlung von PowerPoint-Folien in SVG sind allerdings in einer Reihe von Fällen Probleme mit der Zeichenpositionierung in Verbindung mit Trennstrichen aufgetreten. In durch den SVGmaker erzeugten SVG-Dokumenten werden Textelemente in der in Code-Beispiel 8-1 dargestellten Form repräsentiert. Dabei werden Textblöcke jeweils in eigenen *tspan*-Elementen gekapselt. In diesen Elementen werden die X- und Y-Koordinaten der einzelnen Buchstaben und der eigentliche Text gespeichert. Trennstriche speichert der SVGmaker jeweils in separaten *tspan*-Elementen. In einigen Versionen des SVGmakers wird die X-Position des Trennstriches und der nachfolgenden Zeichen falsch berechnet, was dazu führt, dass die Zeichen übereinander statt nebeneinander positioniert werden.

[78] Es handelt sich hierbei nicht um den schreibgeschützten Modus, in dem die Präsentation lediglich nicht am selben Ort gespeichert werden kann. In diesem Modus ist es nicht möglich, irgendeine Art von Änderungen an der Präsentation vorzunehmen.

```
...
<text class="t1"><tspan x="2206,2253,2284,2315,2346,2357,2373
,2388,2419,2435,2466, 2485,2516,2547" y="2207">Modul
Internet</tspan><tspan x="2563" y="2207">-</tspan><tspan
x="2581,2616,2647,2675,2706,2737,2768,2779,2810,2841,2854,2885"
y="2207">Technologien</tspan></text>
...
```

**Code-Beispiel 8-1: Repräsentation von Text in SVG**

Zur Umgehung dieses Problems kann die Tatsache ausgenutzt werden, dass fehlende Angaben der x-Position von Buchstaben in dem zur Wiedergabe des SVG genutzten ASV so umgesetzt werden, dass die betreffenden Buchstaben automatisch typographisch richtig positioniert werden. In einem Nachbearbeitungsschritt werden daher im Generatordienst alle über den SVGmaker erzeugten SVG-Dokument so manipuliert, dass Angaben zur x-Position von *tspan*-Elementen, die einen Trennstrich enthalten oder aufgrund eines vorgelagerten Trennstriches falsch positioniert sein könnten, gelöscht werden.

Zur Nachbearbeitung wird das SVG-Dokument in C# über ein Objekt der Klasse `System.Xml.XmlTextReader` geladen. Anschließend werden alle *text*-Elemente des Dokumentes einzeln abgearbeitet. Dabei werden alle *tspan*-Elemente des jeweiligen *text*-Elementes analysiert. Eine Analyse über die Grenzen des *text*-Elementes hinweg und damit eine Einbeziehung weiterer *tspan*-Elemente ist nicht nötig, da das Problem nur bei Textbausteinen innerhalb eines *text*-Elementes auftritt. Wird ein Trennstrich gefunden, wird zunächst auf Gleichheit der y-Koordinaten des vorangehenden *tspan*-Elementes und des *tspan*-Elementes, in dem sich der Trennstrich befindet, geprüft. Anschließend wird geprüft, wie groß die Entfernung zwischen der letzten Zeichenposition des vorangehenden *tspan*-Elementes und der Zeichenposition des Trennstriches ist. Liegt der Trennstrich auf der selben y-Position wie die Zeichen und ist die Entfernung des letzten Zeichens des vorangehenden *tspan*-Elementes zum Trennstrich innerhalb eines Toleranzbereiches, der von der verwendeten Schriftgröße abhängt, wird das den Trennstrich beinhaltende *tspan*-Element gelöscht und der Trennstrich wird in das vorangehende *tspan*-Element verschoben. Mit nachfolgenden *tspan*-Elementen wird analog verfahren. Die Beschränkung des Verfahrens auf einen schriftgrößenabhängigen Toleranzbereich dient dabei der Vermeidung von Operationen auf *tspan*-Elementen, die nicht mehr zu einem vorangehenden Trennstrich gehören, sondern, beispielsweise bei mehrspaltigem Text, eigenständige Elemente darstellen.

## 8.3 Indizierung der SVG-Folien zur Volltextsuche

In vielen SVG-Viewern wie dem ASV existiert zwar eine Volltextsuche, es besteht aber keine Möglichkeit, diese Eigenschaft des Viewers scriptgesteuert zu nutzen. Um im Interface dennoch eine Volltextsuche zu realisieren, werden die SVG-Dokumente, die später im Interface als Titelfolien verwendet werden, entsprechend aufbereitet. Dazu wird die Anzahl aller *tspan*-Elemente eines Dokumentes in der Wurzel des Dokumentes gespeichert und alle *tspan*-Elemente werden mit einer eindeutigen ID versehen. Durch diesen Schritt kann der im Dokument enthaltene Text im Interface effizient über das DOM durchsucht werden.

## 8.4 Verlinkung animierter Folienelemente in den Film

Dieser Abschnitt beschreibt die technische Realisierung der bereits in 6.2.2 dargestellten Navigationsmöglichkeiten durch Anklicken bzw. Überfahren von in der Präsentation animierten Folienelementen in der Navigationsansicht. Der Zeitpunkt im Film, zu dem ein

Folienelement verlinkt, wird dabei durch den Zeitpunkt des ersten Erscheinens des entsprechenden Elementes während der Aufzeichnung bestimmt. Das Zeitintervall, das beim Überfahren eines Elementes mit dem Mauszeiger im FootprintBar markiert wird, errechnet sich aus dem Zeitpunkt des ersten Erscheinens des Elementes als Startzeitpunkt und dem Anfangspunkt der darauf folgenden Animation als Endzeitpunkt.
Die eigentliche Verlinkung der in der Präsentation animierten Folienbestandteile wird durch die Einbindungen von JavaScript-Code direkt in das darstellende SVG bewerkstelligt.
Dazu müssen die in der Präsentation animierten Bestandteile einer Folie in der Navigationsansicht zunächst als solche identifiziert werden. Anschließend wird das Element in ein Gruppierungs-Element (<g>) eingeschlossen, dessen *onclick*-, *onmouseover*- und *onmouseout*-Attribute mit den entsprechenden Methodenaufrufen versehen werden.[79] Abb. 8-6 zeigt eine schematische Darstellung dieses Vorgangs. Die Aufrufparameter der Methoden wie Startzeitpunkt und Endzeitpunkt der Animation werden aus der Eventtabelle der Aufzeichnung gelesen und direkt in den Methodenaufruf hineingeschrieben. Das Einfügen der jeweiligen Gruppierungs-Elemente in ein SVG-Dokument gestaltet sich in C# dank der entsprechenden Bibliothek zu Manipulation von XML (`System.Xml`) recht einfach.
Da die Verbindung zwischen Animationen und Folienelementen bei der Konvertierung einer Folie durch den SVGmaker verloren geht, ist die Identifikation animierter Elemente etwas aufwändiger. Dazu wird die Folie so als SVG gedruckt, wie sie später in der Navigationsansicht aussehen soll. In einem parallelen Arbeitsschritt werden alle auf der Folie animierten Bestandteile jeweils einzeln in einer eigens zu diesem Zweck erzeugten Hilfsfolie platziert, die mit dem SVGmaker in SVG umgewandelt wird. Diese parallel erzeugten Folien dienen später als Vergleichsdokumente, die eine Identifikation der animierten Folienbestandteile der Navigationsfolie ermöglichen.

**Abb. 8-6: Schematische Darstellung der Einbettung von JavaScript-Aufrufen in SVG. Zunächst wird ein SVG-Element extrahiert und anschließend in ein mit dem entsprechenden Methodenaufruf versehenes <g>-Element eingebunden.**

Durch Entfernen des weißen Hintergrundes, der im SVG als eigenständiges Objekt repräsentiert wird, ist es dann möglich, die Beschreibung des animierten Elementes zu isolieren. Anschließend werden alle Bestandteile der Navigationsfolie, zu denen ein äquivalentes, d.h.

[79] Eine genauere Beschreibung zur Einarbeitung von JavaScript in SVG lässt sich in [Eisenberg 2002], Kapitel 11 finden.

darstellungsgleiches Element in der Beschreibung des animierten Elementes gefunden wird, mit dem entsprechenden JavaScript-Code versehen. Die Verwendung des Kriteriums Darstellungsgleichheit ist an dieser Stelle nötig, da der SVGmaker, abhängig vom Gesamtbild einer Folie, gleiche Elemente im SVG unterschiedlich beschreibt. Ein reiner Syntax-Vergleich würde daher nicht zu dem gewünschten Ergebnis führen. Im weiteren Verlauf dieses Abschnittes wird die Arbeitsweise des SVGmakers und davon ausgehend die Arbeitsweise des im Rahmen der vorliegenden Arbeit implementierten Codes zur Identifikation von darstellungsgleichen SVG-Elementen behandelt.
Die teilweise sehr ausführliche Diskussion der Struktur des durch den SVGmaker erzeugten SVG in diesem Abschnitt ist dabei beabsichtigt, um die Möglichkeit schneller Anpassungen an Veränderungen in zukünftigen Versionen des SVGmaker zu gewährleisten.

### 8.4.1 Arbeitsweise des SVGmakers

Die Betrachtung des Aufbaus der durch den SVGmaker aus PowerPoint erzeugten SVG-Dokumente ist entscheidend für das Verständnis des im weiteren Verlauf dieses Abschnitts beschriebenen Verfahrens.
Wie bereits in 7.3.1.3 dargestellt, wird der SVGmaker unter Windows als Druckertreiber installiert und kann somit die Druckausgabe beliebiger Programme als SVG speichern. Da PowerPoint in der Druckausgabe keine Optimierungen vornimmt, die nicht-sichtbare Folienbestandteile löschen, werden alle graphischen und damit auch alle verdeckten Bestandteile einer Folie komplett vom SVGmaker in SVG umgewandelt. Daher können auch teilweise verdeckte animierte Elemente gefunden werden. Diese Arbeitsweise der Druckausgabe von PowerPoint wird nicht nur bei der Ansicht von durch den SVGmaker erzeugten SVG-Dateien deutlich. Auch bei der Analyse von Postscript-Dateien, die über die Druckausgabe direkt aus PowerPoint erzeugt werden, treten die beschriebenen Effekte auf. Sehr leicht nachvollziehbar wird die zugrunde liegende Arbeitsweise bei der Ansicht von komplexeren pdf-Dateien, die mit Hilfe des Adobe Acrobat Distillers aus PowerPoint erzeugt worden sind. Im Adobe Acrobat Viewer kann beim Öffnen von Dateien mit genügend hoher Komplexität beobachtet werden, wie sich eine obere Schicht von Objekten während des Aufbaus der Darstellung über eine untere Schicht legt und diese erst nachträglich verdeckt.
Die Tatsache, dass aus PowerPoint erzeugte SVG-Dateien in jedem Fall die komplette Beschreibung aller ihrer graphischen Bestandteile enthalten, hat die Implementation des Vergleichsalgorithmus an vielen Stellen entscheidend vereinfacht. So ist beispielsweise der Vergleich von Objekten immer auf syntaktischer Basis möglich.
In einigen Fällen sind dazu allerdings erhebliche Vorverarbeitungsschritte wie die Transformation relativer in absolute Koordinaten (siehe 8.4.4) oder auch die Sammlung von Zuordnungen bestimmter Werte in Hash-Tabellen (siehe 8.4.6 und 8.4.8) nötig. Der deutlich größere Aufwand einer Neuberechnung der einzelnen Objektbeschreibungen unter Berücksichtigung der vorhandenen Überdeckungen konnte jedoch aufgrund des Vorhandenseins der jeweils kompletten Objektbeschreibungen in der Druckausgabe umgangen werden.

### 8.4.2 Erzeugung der Vergleichsdokumente

Die SVG-Dokumente, die später in der Foliennavigationsleiste des Interfaces zu sehen sind, werden im Rahmen des Zerlegungsprozesses erzeugt (vgl. 8.1.1.2). Im Gegensatz dazu wird zur Erstellung der Vergleichsdokumente ein leicht abgewandelter Algorithmus benötigt. Beide Algorithmen zerlegen die Folie in Einzelschritte, die jeweils den Zustand der Folie nach Eintreten jeder Folienanimation[80] darstellen. Im Gegensatz zum Zerlegungsalgorithmus, der jeweils die Gesamtansicht einer Folie direkt nach dem Abschluss der betreffenden

[80] Parallel eintretende Effekte sind dabei als eine zusammenhängende Animation zu verstehen.

Animation herstellt, müssen im Vergleichsdokument jedoch nur genau die Elemente enthalten sein, die mit der entsprechenden Animation auf der Folie erschienen sind. Die Reihenfolge der Abarbeitung sowie die zugrunde liegende Logik sind dabei in beiden Algorithmen gleich. Unterschiede existieren lediglich in der Erstellung der PowerPoint-Folien, die zur Erzeugung der SVG-Dokumente genutzt werden. Zur Vereinfachung und Effizienzsteigerung des Codes werden die entsprechenden Methoden der zur Verlinkung der SVG-Navigationsfolien programmierten Klasse *Linker.cs* daher aus der Aktualisierungsmethode des Zerlegungsalgorithmus heraus aufgerufen.
Die Erzeugung der Vergleichsfolien ähnelt zwar in ihrer grundlegenden Konzeption den Methoden zum Folienaufbau im Zerlegungsalgorithmus, konnte jedoch aufgrund der weitaus geringeren Anforderungen an die Struktur der Vergleichsdokumente merklich laufzeiteffizienter gestaltet werden. Ein Grund hierfür ist der Wegfall der in 8.1.1.2.2 beschriebenen Z-Order-Problematik, da der Vergleich der einzelnen Knoten ohne Bezug auf deren Position im SVG-Dokument möglich ist und da auch verdeckte Elemente vollständig in den erzeugten SVG-Dokumenten enthalten sind. Eine weitere Vereinfachung ergibt sich aus der Tatsache, dass Textbausteine unabhängig von der umgebenden Textbox betrachtet werden können. Dabei auftretende, optisch unschöne Effekte können vernachlässigt werden, da sie für den Vergleich keine Rolle spielen.
Dem Wegfall einiger Probleme aus der Implementation des Zerlegungsalgorithmus stehen jedoch auch zusätzliche Anforderungen gegenüber. Aufgrund der Tatsache, dass auch auf der Folie verdeckte Elemente mit in das SVG-Dokument übernommen werden, kann die Erzeugung der Vergleichsdokumente nicht einfach durch Drucken einer modifizierten Version der Ursprungsfolie realisiert werden. Auch wenn in der Ursprungspräsentation alle nicht benötigten Elemente von einer Folie entfernt werden, sind auf der Folie in den meisten Fällen immer noch Seitenzahlen und eventuell auf der Masterfolie der Präsentation enthaltene Formen und Bilder vorhanden. Veränderungen der Masterfolie bergen die Gefahr unbeabsichtigter Auswirkungen auf die Folien der Präsentation. Auch die Erzeugung der Vergleichsdokumente in einer zweiten Präsentation ohne die genannten Folienelemente wie Seitenzahlen oder Formen und Bilder der Masterfolie ist ausgeschlossen. In diesem Fall tritt die bereits in 8.1.1.1.1 erwähnte Problematik der Veränderung von Voreinstellungen wie Schriftgrößen und Farbschemata auf. Das Vergleichsdokument wird daher in mehreren Schritten erzeugt. Zunächst wird ein Duplikat der Ursprungsfolie erzeugt, auf dem alle Elemente bis auf die zum jeweiligen Animationsschritt gehörigen Elemente gelöscht werden. Der Folienhintergrund bleibt dabei unverändert. Anschließend wird die Folie in ein SVG-Dokument, das Referenzdokument, umgewandelt. In einem weiteren Schritt werden auch alle zum Animationsschritt gehörigen Effekte auf der Folie gelöscht und die Folie wird in ein weiteres SVG-Dokument, das Hintergrunddokument, umgewandelt. Im letzen Schritt werden unter Nutzung des im weiteren Verlauf beschriebenen Vergleichsalgorithmus' alle Elemente aus dem Referenzdokument gelöscht, die auch im Hintergrunddokument enthalten sind. Der Vergleichsalgorithmus wird dabei in gleicher Form zur Verlinkung animierter Folienelemente und zur Löschung von Folienhintergrundelementen genutzt. Auf der Ebene der Programmierung ist dies durch die Verwendung eines Delegate-Patterns [Gamma, Helm, Johnson & Vlissides 1995] realisiert. Dabei werden dem Vergleichsalgorithmus zur Erledigung der jeweiligen Aufgabe jeweils spezialisierte Objekte übergeben. Da die Objekte die gleiche Schnittstelle implementieren, können sie ohne Abänderung des Programmcodes, der den Vergleichsalgorithmus beschreibt, genutzt werden.
Auch die Problematik gleichzeitig auftretender Effekte (vgl. 8.1.1.2.2) ist bei der Erstellung der Vergleichsdokumente zu berücksichtigen. Aus diesem Grund werden alle während einer Animation auf der Folie erscheinenden Objekte auf die entsprechende Vergleichsfolie kopiert. Dank der in 8.1.1.2.1 beschriebenen Konvertierung schließt dieses Vorgehen auch

die Animationen Pfad und Größenänderung (siehe 8.1.1.2.1.1 und 8.1.1.2.1.2) mit ein. Nach jedem Effekt, der nicht gleichzeitig mit dem vorangehenden Effekt stattfindet, wird die Vergleichsfolie als SVG ausgedruckt. Dies passiert insbesondere auch dann, wenn der aktuelle Effekt ein Objekt von der Folie entfernt hat, aber im selben Zeitschritt erfolgte Effekte andere Objekte hinzugefügt haben.

### 8.4.3 Grundlegender Ablauf des Vergleichsalgorithmus

Der Vergleichsalgorithmus sucht für jedes graphische Element des Originaldokumentes ein äquivalentes graphisches Element im Vergleichsdokument. Nicht-graphische Elemente, wie Gruppierungs-Elemente oder auch verschachtelte SVG-Elemente sind dabei aufgrund der Tatsache, dass sie nur der inhaltlichen Organisation der SVG-Dokumente dienen, nicht relevant und können übersprungen werden. Da der SVG-Baum des Vergleichsdokumentes für jedes graphische Element des Original-Dokumentes jedoch im worst case einmal durchlaufen werden muss, können verschachtelte Gruppierungs- und SVG-Elemente die Laufzeit des Algorithmus erheblich beeinflussen. Zur Vereinfachung der Suche und damit zur Verringerung der Laufzeit werden daher alle verschachtelten Gruppierungs- und SVG-Elemente im Vergleichsdokument in einem Vorverarbeitungsschritt gelöscht. Ist $m$ dabei die Zahl der graphischen Elemente im Originaldokument, $n$ die Zahl der graphischen Elemente im Vergleichsdokument und $k$ die Zahl der Gruppierungs- und SVG-Elemente sinkt die Laufzeit in O-Notation damit von dem in Formel 8-2 gezeigten Wert

$$O(m(n+k))$$

**Formel 8-2: Laufzeit des Vergleichsalgorithmus ohne Optimierung**

auf den in Formel 8-3 gezeigten Wert.

$$O(mn)$$

**Formel 8-3: Laufzeit des Vergleichsalgorithmus bei destruktiver Arbeitsweise im Vergleichsdokument**

Nach dieser Vorverarbeitung ist das Vergleichsdokument ein Baum mit $n$ Blättern, die direkt unter der Wurzel angeordnet sind. Da keine Zwischenebenen vorhanden sind, kann das Vergleichsdokument damit als Liste betrachtet werden. Beim Originaldokument hingegen handelt es sich weiterhin um einen Baum, der vom Algorithmus rekursiv durchlaufen wird.
Das Auflösen der Struktur des Vergleichsdokumentes ist hinsichtlich der Darstellbarkeit des Dokumentes destruktiv. Da das Vergleichsdokument jedoch nur zur Identifikation animierter Elemente benötigt wird, stellt dieser Sachverhalt kein Argument gegen die hier vorgestellte Vorgehensweise dar. Eine weitere Laufzeitverringerung ließe sich durch die Verwendung von Hashtabellen zum Zugriff auf die Elemente des Vergleichsdokumentes realisieren, wie beispielsweise bei herkömmlichen Implementationen zur Navigation mit XPATH gegeben [Chan, Felber, Garofalakis & Rastogi 2001].

### 8.4.4 Repräsentation gleichfarbiger Polygone und Linien im SVG

Obwohl in SVG spezielle Tags zur Beschreibung von Linien und Polygonen existieren, ist es auch möglich, sowohl Linien als auch Polygone durch *path*-Tags zu beschreiben [47].
Dabei werden von einer aus Geradensegmenten bestehenden Linie eingeschlossene Flächen als Fläche eines Polygons interpretiert. Die Polygone werden jeweils in der im *style*-Attribut als *fill* angegebenen Farbe dargestellt (vgl. Code-Beispiel 8-2).[81]

[81] Da der SVGmaker ausschließlich die genannte Beschreibung von Linien und Polygonen durch *path*-Tags verwendet, wird der Begriff Pfad im Folgenden synonym zu den Begriffen Linie und Polygon verwendet.

In SVG besteht zudem die Möglichkeit nicht miteinander verbundene Linien in ein und demselben *path*-Tag zu beschreiben. So zeigt Code-Beispiel 8-2 SVG, das in einem beliebigen SVG-Viewer als drei voneinander getrennte Pfeile dargestellt wird. Dabei werden nur die Umrisse der Pfeile beschrieben, die eingegrenzte Fläche wird automatisch vom Renderer berechnet. Im SVGmaker wird diese Möglichkeit der Codierung genutzt, um die Größe der erzeugten SVG-Dateien gering zu halten.[82]
Wenn nicht alle Linien oder Polygone, die in der Navigationsfolie in einem *path*-Tag zusammengefasst werden, auch Bestandteil derselben Animation sind, kann eine animierte Linie also durch einfachen Syntax-Vergleich der *path*-Tags beider SVG-Dokument-Bäume nicht gefunden werden. Um einen Vergleich der *path*-Tags beider Bäume dennoch zu ermöglichen, werden die *path*-Tags beider Bäume zu je einem Tag pro beschriebener Linie zerlegt.
Zum Verständnis dieser Zerlegung ist zunächst eine genauere Betrachtung des Code-Fragments in Code-Beispiel 8-2 nötig. Die in der Abbildung jeweils rot markierte Zeichenfolge „z m“ bezeichnet in einer SVG-Pfadbeschreibung die Grenze zwischen zwei Einzelpfaden [47]. Das Kommando „z“ bedeutet dabei *closepath* und zieht eine gerade Linie zum Ausgangspunkt der vorangegangenen Zeichenkommandos, „m“ bedeutet *moveto* und leitet das Zeichnen eines neuen Pfades durch das Setzen eines neuen Ausgangspunktes ein. Großschreibung bezeichnet ein *moveto* in Absolutkoordinaten, Kleinschreibung ein *moveto* in Relativkoordinaten. Das Kommando „l“ steht für *lineto* und initiiert das Zeichnen eines Pfades.

```
<path style="fill:#666699;stroke:none" d="M 521,1567 l 64,0
0,298 128,0 0,-298 64,0 -128,-100 -128,100 z m 429,-1 l 64,0
0,298 128,0 0,-298 64,0 -128,-100 -128,100 z m 1736,-316 l
64,0 0,298 128,0 0,-298 64,0 -128,-100 -128,100 z"/>
```

**Code-Beispiel 8-2: Der SVGmaker zeichnet unterschiedliche Objekte als nicht zusammenhängende Pfade in ein und demselben *path*-Tag.**

Bei der Zerlegung in Einzelpfade ist die Verwendung relativer Pfadangaben durch den SVGmaker zu berücksichtigen. Die beiden Startkoordinaten hinter einem relativen *moveto* müssen bei der Umwandlung in neue Pfad-Objekte in absolute Koordinaten umgerechnet werden, die dann als Startkoordinaten der jeweiligen Linien in den neu erzeugten Pfad-Objekten dienen können.
Bei **Polygonen** wie in Code-Beispiel 8-2 ist dies ohne größeren Aufwand möglich, da durch das *closepath* am Ende jedes Einzelpfades jeweils eine Linie zurück zum Anfang des Zeichenkommandos gezogen wird. Die Anfangskoordinaten beim Zeichnen einer geometrischen Form sind daher mit den Endkoordinaten identisch. Bei der Zerlegung eines zusammengesetzten Pfadobjektes in Einzelpfade bedeutet dies, dass zur Berechnung der Anfangskoordinaten der jeweiligen Einzelpfade jeweils nur die Anfangskoordinaten des vorangehenden Pfadobjektes bekannt sein müssen. Eine Ausnahme bilden hier Rechtecke. Diese werden durch das Tag <rect> gekennzeichnet und durch ein X-Y-Koordinatenpaar sowie Höhe und Breite beschrieben. Der Vergleich dieser Elemente lässt sich aufgrund der Einfachheit ihrer Beschreibung rein syntaktisch realisieren.

[82] Rechtecke stellen hier eine Ausnahme dar. Diese werden vom SVGmaker in *rect*-Elementen beschrieben. Da der Vergleich dieser Elemente sehr einfach durchgeführt werden kann, wird er hier nur am Rande erwähnt.

Stellt ein Pfadobjekt dagegen eine Ansammlung von **nicht miteinander verbundenen Linien** dar, ist der Endpunkt einer Linie nicht mit dem Startpunkt der folgenden Linie identisch. In diesem Fall muss zur Berechnung der Anfangskoordinaten einer Linie die gesamte Beschreibung der vorhergehenden Linie geparst werden. Das Parsen der Linien selbst gestaltet sich vergleichsweise einfach, da die Struktur der zu erwartenden Beschreibungen durch die Spezifikation des SVG-Formates vorgegeben und auf ein sehr kleines Vokabular beschränkt ist. Die aufeinander folgenden Koordinatenangaben können dabei mittels einfacher String-Operationen extrahiert und dann in Werte vom Typ Double umgewandelt werden. Abschließend werden anhand dieser Werte die absoluten Startkoordinaten des nachfolgenden Pfades berechnet.

### 8.4.5 Repräsentation von Kurven und Ellipsen im SVG

Obwohl in SVG bereits seit der Version 1.0 Tags und Kommandos zur Beschreibung von Kurven und Ellipsen wie beispielsweise das *ellipse*-Tag und das Kommando *curveto* im *path*-Tag existieren, exportiert der SVGmaker Kurven und Ellipsen als eine zusammenhängende Anreihung von Liniensegmenten. Beschrieben werden diese Liniensegmente in *path*-Tags. Probleme, die beim Vergleich dieser Formen auftreten könnten, sind also durch die Reduktion auf Probleme, die beim Vergleich von Linien und Polygonen auftreten können, bereits gelöst.
Diese Darstellung in Form von Liniensegmenten ist übrigens nicht in der Druckausgabe von PowerPoint begründet. Laut [Adobe Systems Incorporated 1999] lässt PostScript die Beschreibung von Kurven durch das Kommando *curveto*[83] zu. Die genauere Betrachtung einer PostScript-Datei, die durch Umleitung der für einen PostScript-Drucker bestimmten Ausgabe in eine Datei entstanden ist, zeigt dann, dass Kurven aus der PowerPoint-Präsentation auch im PostScript als Kurven codiert werden.
Bei Windows-Rechnern, an die kein PostScript-Drucker angeschlossen ist, lässt sich dieser Effekt leicht durch die Installation eines generischen PostScript-Druckertreibers, wie beispielsweise kostenlos von Adobe unter [28] zum Download bereitgestellt, nachvollziehen.

### 8.4.6 Repräsentation von Bildern im SVG

Bilder werden vom SVGmaker nicht direkt in das erstellte Dokument, sondern zunächst in *image*-Tags eingebunden, die später von *use*-Tags aus per *xlink* referenziert werden. Die *image*-Elemente sind ihrerseits in *defs*-Elemente eingebettet. Obwohl die Spezifikation des SVG-Formates dies prinzipiell ermöglicht [47], verfügen die *use*-Tags dabei nicht über eigene Koordinaten, sondern werden mit Hilfe eines umgebenden *svg*-Tags positioniert.
Diese Art der Codierung lässt sich an dieser Stelle vor allem dadurch erklären, dass so bei mehrfacher Verwendung desselben Bildes die Größe der betreffenden SVG-Datei klein gehalten werden kann.
Da die jeweiligen *image*-Elemente über im Dokument eindeutige IDs verfügen, diese IDs aber bei zwei unterschiedlichen Dokumenten auf unterschiedliche *image*-Elemente verweisen können, werden zunächst in einem Vorverarbeitungsschritt Hashtabellen mit den Zuordnungen der IDs beider Dokumente zu den jeweils referenzierten *image*-Elementen erzeugt.
Ein zusätzlicher Vorverarbeitungsschritt ist aufgrund der Positionierung der *use*-Elemente durch das jeweils umgebende *svg*-Element in Verbindung mit der in 8.4.3 beschriebenen Löschung von nicht-graphischen Elementen nötig. Da beim Vergleich von zwei *use*-Elementen auch deren Position auf der Folie berücksichtigt werden muss, werden die Koordi-

[83] SVG basierte in der Anfangsphase seiner Entwicklung auf Postscript [Watt 2001]. Die Namensgleichheit einiger Elemente sowie auch strukturelle und konzeptionelle Ähnlichkeiten sind daher nicht zufällig.

natenangaben des umgebenden *svg*-Elementes kurz vor dessen Löschung in das eingeschlossene *use*-Element kopiert.
Zum Vergleich von zwei *use*-Elementen können dann die Positionsangaben des umgebenden SVG-Elementes des *use*-Elementes des Originaldokumentes und die Positionsangaben des *use*-Elementes des Vergleichsdokumentes herangezogen werden. Der Vergleich der referenzierten Teilbilder ist durch Verwendung der Hashtabellen möglich.
Eine weitere potentielle Problemquelle ist die Zerlegung von Bildern aus der PowerPoint-Präsentation in mehrere Teilbilder im erzeugten SVG durch den SVGmaker. Eine größere Testserie von Konvertierungen hat jedoch gezeigt, dass gleiche Bilder auch in unterschiedlichen Kontexten immer in gleicher Art und Weise zerlegt werden, so dass an dieser Stelle zumindest bei den bisher verwendeten Versionen des SVGmaker kein Handlungsbedarf besteht. Einen Sonderfall stellen Bilder mit transparentem Hintergrund dar. Aufgrund seiner Komplexität wird dieses Thema im folgenden Unterabschnitt gesondert behandelt.

### 8.4.7 Verarbeitung von Bildern mit transparentem Hintergrund durch den SVGmaker

Bilder mit transparentem Hintergrund werden vom SVGmaker in eine Vielzahl von Teilbildern zerlegt. Abhängig von der Größe der betreffenden Bilder in der PowerPoint-Präsentation kann ein Bild daher durch **mehrere Tausend Knoten** im SVG-Dokument dargestellt werden. Aufgrund der in 8.4.6 beschriebenen Arbeitsweise des SVGmakers, Bilder nicht direkt, sondern nur über Verweise darzustellen, wird die Anzahl der Knoten sogar noch einmal verdoppelt. Fälle, in denen ein Bild in mehrere Tausend Knoten zerlegt wird, treten in der Realität nur sehr selten auf. In Anbetracht einer durchschnittlichen Größe der erzeugten SVG-Dokumente in einer Größenordnung von unter hundert Knoten, bedeutet dies jedoch eine enorme Vervielfachung der Knotenanzahl der Dokumente. Aufgrund der Tatsache, dass ein Bild sowohl im Vergleichsdokument als auch im Originaldokument vorkommen kann und der in 8.4.3 hergeleiteten Laufzeit eines Folienvergleiches (siehe Formel 8-3) können daher beim Vergleich zweier Folien nicht zumutbare Verarbeitungsdauern auftreten.
Um dieses Problem zu umgehen, ist eine Reihe von Lösungsansätzen denkbar. Im Folgenden werden diese Ansätze diskutiert, um die Wahl des implementierten Ansatzes zu erläutern.
Der **Austausch des jeweiligen Bildes** in der PowerPoint-Präsentation durch das selbe Bild **in einem anderen Grafikformat** ließe sich ohne größere Probleme realisieren. Da hier allerdings ein Grafikformat mit nicht transparentem Hintergrund gewählt werden müsste, wäre eine Darstellungsgleichheit im Gesamtkontext der betroffenen Folie nicht gegeben. Dieser Ansatz wird daher als suboptimal verworfen.
Eine **Umstrukturierung des Vergleichsalgorithmus zur Verwendung von Hashtabellen** ist wenn überhaupt nur mit sehr großem Aufwand machbar, da der beschriebene Vergleichsalgorithmus nicht auf Objekt-, sondern auf Darstellungsgleichheit der Elemente basiert. Im Vergleich mit der letztendlich vorgestellten Lösung sind zudem keine Laufzeitgewinne zu erwarten, die dieses Vorgehen rechtfertigen würden.
Die **Löschung bereits gefundener Elemente im Vergleichsdokument** stellt eine recht einfache Möglichkeit zur Verringerung der Laufzeit dar. Die Laufzeit eines Dokumentenvergleiches würde damit halbiert werden. Dieses Ergebnis ergibt sich aus der Berechnung einer Teleskopsumme. Dabei werden dem ersten Schleifendurchlauf beim Vergleich mit $n$ Elementen der letzte Schleifendurchlauf beim Vergleich mit 0 Elementen, dem zweiten Schleifendurchlauf beim Vergleich mit $n$-1 Elementen der vorletzte Schleifendurchlauf beim Vergleich mit einem Element usw. gegenübergestellt. Für die Laufzeit des Vergleichs in O-Notation ist eine solche Halbierung natürlich nicht relevant. Auch für den praktischen Ein-

satz führt diese Laufzeitverringerung angesichts einer Verarbeitungsdauer von Tagen nicht zu einem akzeptablen Ergebnis.
Eine **zeitliche Begrenzung des Vergleichs** zweier Folien auf einen festen Zeitraum ist zwar bezüglich der Reallaufzeit effektiv, führt jedoch zu einer nicht korrekten Arbeitsweise des Vergleichsalgorithmus', da der Vergleich vorzeitig abgebrochen wird. Resultat eines derartigen Vorgehens wären daher nur zu einem Bruchteil mit Links versehene Folien.
Aufgrund der beschriebenen Unzulänglichkeiten der vorgestellten Alternativen wurde für die Implementation die **Externalisierung der Bilder** in einer separaten Bilddatei gewählt. Dazu wird zunächst wieder die COM-Schnittstelle von PowerPoint genutzt. Als erster Schritt wird das jeweilige Bild in eine leere PowerPoint-Präsentation kopiert. Diese Präsentation wird als HTML exportiert, wobei eingebettete Bilder als eigene Dateien gespeichert werden. Der Speicherort der Bilddatei kann anschließend aus der von PowerPoint beim Export erzeugten Datei *filelist.xml* herausgelesen werden. Nachdem die Datei aus dem von PowerPoint als HTML-Export erzeugten Ordner heraus kopiert worden ist, muss ein Verweis auf die Bild-Datei in das SVG-Dokument eingefügt werden und die nicht mehr benötigten Bild-Knoten und Bild-Definitionen müssen aus dem Dokument gelöscht werden. Ein Verweis auf ein externes Bild ist in Code-Beispiel 8-3 dargestellt.
Zur Erzeugung dieses Verweises müssen zunächst x- und y-Koordinaten sowie die Höhe und Breite des Bildes im SVG ermittelt werden. Dazu wird die PowerPoint-Präsentation, die bereits zum HTML-Export genutzt wurde mit dem SVGmaker in SVG umgewandelt. Da in dieser Präsentation außer dem Bild keine anderen graphischen Objekte enthalten sind, dienen alle vorhandenen Elemente der Darstellung des Bildes. Da das Bild sowohl entlang der x-Achse als auch entlang der y-Achse fragmentiert sein kann, werden alle Knoten des Dokumentes durchlaufen, um den minimalen x-Wert und den minimalen y-Wert zu ermitteln. Im selben Durchlauf wird die Höhe und Breite des Bildes aus den Daten der Bildfragmente berechnet. Um später das Element identifizieren zu können, das den Anfangspunkt des Bildes repräsentiert, wird zusätzlich immer der Knoten gespeichert, der das Bildfragment mit den niedrigsten x- und y-Werten repräsentiert.[84]

```
<image preserveAspectRatio="none" x="306" y="483"
width="1780" height="621"
xlink:href="externalPictures\picture_11.jpg" />
```

**Code-Beispiel 8-3: Verweis auf ein externes Bild aus einem SVG-Dokument heraus**

Der nächste Schritt besteht darin, den Verweis auf das externe Bild in das SVG-Dokument einzufügen und die dadurch unnötig gewordenen Verweise und Bildfragmentdefinitionen zu entfernen. Da in SVG die Position von Objekten innerhalb der z-Ebene durch die Position der beschreibenden Elemente im SVG-Baum bestimmt wird [Watt & Lilley 2002], wird das Verweiselement an die Stelle des Elementes im SVG-Baum eingefügt, das den Anfangspunkt des Bildes repräsentiert.
Dazu wird zunächst in dem SVG-Dokument, das das eigentliche Konvertierungsresultat darstellt, der Knoten gesucht, der mit dem Anfangselement identisch ist, also das Anfang-

[84] Alternativ zu dieser Vorgehensweise wäre auch eine dynamische Berechnung der x- und y-Koordinaten des Bildanfangs aus den Bilddaten der PowerPoint-Präsentation möglich. Ein solches Verfahren hätte jedoch den Nachteil, dass die zu verwendende Transformation mit Einstellung und Version des Drucktreibers variieren kann. Außerdem wäre es bei einer solchen Lösung nicht möglich, festzustellen, ob das referenzierte Bildfragment auch tatsächlich Teil des zu ersetzenden Bildes ist.

selement im Konvertierungsresultat darstellt. Sind in diesem Dokument mehrere Bilder enthalten, werden zunächst alle Bilder separat konvertiert, so dass die entsprechenden Anfangselemente der Bilder parallel gesucht werden können. Dadurch ist die Laufzeit des Ersetzungsalgorithmus nur von der Größe der Bilder, nicht aber von deren Anzahl abhängig. Steht $n$ für die Anzahl der Knoten im Dokument vor Ersetzen der Bildknoten, $l$ für die Anzahl der Bilder und $m_k$ für die Anzahl der Knoten in Bild $k$, lässt sich die asymptotische Laufzeit des Ersetzungsalgorithmus daher durch Formel 8-4 bestimmen.

$$O(n + \sum_{k=1}^{l} m_k)$$

**Formel 8-4: Laufzeit des Ersetzungsalgorithmus für Bilder mit transparentem Hintergrund in SVG-Dokumenten.**

Da Bilder vom SVGmaker immer in gleicher Art und Weise umgesetzt werden, ist die Anzahl der Bildfragmente, die zur Darstellung eines Bildes in verschiedenen SVG-Dokumenten erzeugt werden immer gleich. Die Summe aller Knoten zur Darstellung der Bilder, die im eigentlichen Dokument enthalten sind, kann daher nicht größer sein, als die Anzahl der Knoten im Dokument. Der Summenteil in Formel 8-4 kann daher $n$ nicht übersteigen. Als Obergrenze für die Laufzeit des Ersetzungsalgorithmus ergibt sich damit Formel 8-5. Aufgrund der hohen Knotenanzahl in den betroffenen Dokumenten ist die reale Laufzeit jedoch merklich größer als die Laufzeit des in diesem Abschnitt beschriebenen Vergleichsalgorithmus.

$$O(n)$$

**Formel 8-5: Vereinfachte Laufzeitabschätzung für den Ersetzungsalgorithmus.**

Nachdem die Verweise auf die externen Bilddateien in das Dokument eingefügt worden sind, werden alle Definitionen der nun obsolet gewordenen Bildfragmente sowie die Verweise auf diese Definitionen aus dem Dokument entfernt. Auch textuelle Zwischenräume zwischen den entfernten Knoten, die als Knoten des Typs *significant Whitespace* (*#significant-whitespace*) im Dokument vorhanden sind, werden gelöscht. Auch dieser Schritt bedeutet in kritischen Fällen eine Verkleinerung des Dokumentes um mehrere tausend Knoten.

In der regulären Version des virtPresenter wird der beschriebene Code nicht ausgeführt, da Bilder im kritischen Umfang sehr selten sind. Hinzu kommt die Tatsache, dass ausgelagerte Bilder nicht in der in 8.5 geschilderten Form komprimiert werden können und so die Downloadzeiten der erzeugten SVG-Dokumente deutlich erhöhen würden. Der beschriebene Code ist daher nur in einer Testversion implementiert.

### 8.4.8 Repräsentation von Textbausteinen im SVG

Der SVGmaker schreibt Text in der im unteren Teil von Code-Beispiel 8-4 ersichtlichen Form in die von ihm generierten SVG-Dokumente. Das *class*-Attribut eines *text*-Elementes verweist auf einen Style-Eintrag in einem internen CSS, das vom SVGmaker in das erstellte SVG-Dokument integriert wird. Die Verwendung des *class*-Attributes ermöglicht es dabei, Eigenschaften von Textelementen einmal im Dokument zu definieren und mehrmals zu verwenden.[85] Der innere Aufbau eines solchen internen CSS ist im oberen Teil von Code-Beispiel 8-4 zu sehen.

[85] Eine genauere Beschreibung der Verwendung des class-Attributes ist in [47], Abschnitt 6.12 zu finden.

Das Stylesheet enthält dabei Informationen über die im Dokument verwendeten Text-Fonts. Da im Vergleichsdokument nicht alle im Ursprungsdokument verwendeten Textbausteine vorkommen, sind auch die Definitionen der Texteigenschaften in den beiden Dokumenten unterschiedlich. So kann text.t7 aus dem Ursprungsdokument beispielsweise im Vergleichsdokument mit text.t3 benannt sein.

```
text.t3 {font-family:'Tahoma',sans-serif;font-size:112;fill:#FFFFFF}
text.t5 {font-family:'Arial Black',sans-serif;font-size:48;fill:#FFFFFF}
text.t6 {font-family:'Arial',sans-serif;font-size:36;fill:#FFFFFF}
text.t7 {font-family:'Arial',sans-serif;font-size:48;fill:#000000}
...
<text class="t3"><tspan x="275,342,403,443,480,539,565,591"
y="369">Vorteile</tspan></text>
```

**Code-Beispiel 8-4: Textfonts werden vom SVGmaker in einem eingebetteten CSS vordefiniert und anschließend per class-Attribut referenziert.**

Beim Vergleich von zwei *text*-Elementen reicht ein rein syntaktischer Vergleich der *class*-Attribute daher nicht aus. Um diesem Problem zu begegnen, werden daher in einem Vorverarbeitungsschritt zwei Hashtabellen mit den Zuordnungen der Font-Beschreibungen der Stylesheets beider Dokumente angelegt. Da die Stylesheets in einer CDATA-Sektion gekapselt sind, können sie allerdings nicht durch einen XML-Prozessor ausgelesen werden. Zur Extraktion der Font-Beschreibungen aus den Stylesheets werden stattdessen reguläre Ausdrücke verwendet.[86] Die Einträge der beiden Hashtabellen können anschließend syntaktisch verglichen werden.
Ein weiterer Vorverarbeitungsschritt ist nötig, da der eigentliche Text nicht direkt in *text*-Elementen, sondern in eingeschlossenen *tspan*-Elementen enthalten ist. Das Vorkommen mehrerer *tspan*-Elemente in einem *text*-Element ist dabei der Regelfall. Innerhalb der *text*-Elemente können keine Gruppierungselemente eingefügt werden, einzelne *tspan*-Elemente können also nicht animiert werden. Die *text*-Elemente beider Dokumente werden daher so zerlegt, dass jedes *tspan*-Element in einem eigenen *text*-Element eingeschlossen wird. Verarbeitungsprozesse auf Ebene der verwendeten Koordinaten wie etwa in 8.4.6 beschrieben, sind nicht nötig, da in *tspan*-Elementen absolute Koordinatenangaben verwendet werden.
Ein einfacher syntaktischer Vergleich des von zwei *text*-Elementen eingeschlossenen XML ist jedoch nicht möglich aufgrund der Tatsache, dass gleiche Textbausteine unterschiedlich gestückelt in *tspan*-Elemente eingeschlossen werden können. Dieser Effekt tritt beispielsweise bei der Verwendung von Masterfolien im Zusammenhang mit der in diesem Fall nötigen Rücksetzung von Texteigenschaften wie Größe oder Farbe auf (vgl. 8.4.2). Beim Vergleich von *tspan*-Elementen wird daher auch auf syntaktische Gleichheit von Teilabschnitten des eingeschlossenen Textes und der zugehörigen Koordinaten geprüft. Die Anzahl der gleichen Buchstaben im Text wird dabei auch mit der Anzahl der gleichen Koordinaten verglichen, um zufällige Ähnlichkeiten auszuschließen. Gegebenenfalls wird ein *tspan*-Element dann so in mehrere kleinere *tspan*-Elemente aufgeteilt, dass darstellungsgleiche Elemente auch syntaktisch gleich sind. Ein Problem bei dieser Aufteilung stellen Leerzeichen dar. Da bei einer einfachen Zerteilung eines *tspan*-Elementes der rechte Teil des Textes unter Umständen mit einem Leerzeichen beginnt, kann durch die Teilung ein neues *tspan*-Element entstehen, dessen Text mit Leerzeichen beginnt und dessen erste Koordinatenangaben die Positionen dieser Leerzeichen bestimmen. Bei der Erzeugung der Vergleichsdoku-

[86] Eine gute Einführung und ein kompletter Überblick über reguläre Ausdrücke ist in [Friedl 1998] zu finden.

mente mit dem SVGmaker werden jedoch keine Leerzeichen am Anfang eines *tspan*-Elementes positioniert. Daher erschweren die durch die Teilung am Anfang eines Elementes positionierten Leerzeichen den Vergleich. Um dieses Problem zu vermeiden, werden Leerzeichen am Anfang eines durch Teilung erzeugten *tspan*-Elementes und die zugehörigen Koordinatenangaben direkt bei der Aufteilung gelöscht.

### 8.4.9 Schlussbetrachtung

Der SVGmaker ist so konzipiert, dass er die Größe der erzeugten SVG-Dateien möglichst klein hält.[87] So ist beispielsweise die Komprimierung der Dateien mit dem Zip-Algorithmus eine Voreinstellung bei der Konvertierung durch den SVGmaker. Auch die Verwendung von internen Stylesheets und *def*-Elementen sowie die Zusammenfassung von Polygonen können zur Reduzierung der Dateigröße von SVG-Dokumenten genutzt werden [Kunze, Mertens & Vornberger 2005]. Im Gegensatz zum Einsatz des Zip-Algorithmus verwendet der SVG-maker diese Komprimierungsstrategien jedoch in jedem Fall. Einige dieser Komprimierungsstrategien erschweren, wie oben beschrieben, den Vergleich von Dokument-Elementen.
Da es sich bei der hier vorgestellten Technik jedoch um einen vergleichsweise ungewöhnlichen Anwendungskontext handelt, ist die Motivation zur Verwendung von *use*-Tags und eingebetteten CSS mehr als nachvollziehbar. Die problemorientierte Beschreibungsweise dieser Eigenschaften im vorliegenden Abschnitt ist deshalb rein deskriptiv zu verstehen.
Die meisten der implementierten Vorverarbeitungsschritte erhalten den kompakten Aufbau der Dokumente, lediglich bei der Zerlegung von *path*- und *text*-Elementen erhöht sich die Größe der Dokumente. Eine spätere Zusammenlegung nicht-animierter Elemente dieser beiden Kategorien wäre prinzipiell machbar, erscheint aber angesichts der geringen Größenveränderungen durch die in diesem Abschnitt beschriebenen Algorithmen nicht sinnvoll.

## 8.5 Zipping des SVG

Beim Laden des Interfaces über eine Web-Verbindung wird die Wartezeit bei der Initialisierung des Interfaces durch die Größe der verwendeten SVG-Dateien beeinflusst. Um diese Wartezeiten zu minimieren, ist es daher sinnvoll, die Größe der erzeugten Dokumente gering zu halten.
Einen sehr effektiven Weg zur Reduzierung der Größe von SVG-Dateien stellt die Verwendung des Komprimierungsverfahrens ZIP dar [Shigeru & Masaaki 2002]. SVG-Dateien können programmgesteuert mit GZIP komprimiert werden [Adcock 2004]. Viewer wie der ASV dekomprimieren die Dateien automatisch beim Laden. Der Unterschied zwischen GZIP und ZIP besteht lediglich darin, dass ZIP mehrere Dateien in einer ZIP-Datei zusammenfasst und zusätzliche Header Informationen in dieser Datei enthalten sind. GZIP kann jeweils nur eine Datei komprimieren und es wird kein Header erzeugt. Die zur Kompression verwendete Bibliothek ist Open Source und kann unter [33] herunter geladen werden. Die Nutzung der ZIP-Kompression des SVGmaker ist an dieser Stelle nicht möglich, da in diesem Fall die in diesem Kapitel beschriebenen Nachbearbeitungsschritte nicht durchgeführt werden könnten.

[87] Eine Ausnahme stellen hier in das SVG eingebundene Bilder dar, die mit sehr hoher Auflösung eingefügt werden und damit zu sehr großen Dateien führen können.

# 9 Zusammenfassung, Diskussion und Ausblick

In diesem Kapitel werden zunächst kurz in 9.1 die Ergebnisse der Arbeit zusammengefasst. Daran anschließend werden in 9.2 Einsatzerfahrungen und Auswertungen einer Reihe von Nutzerbefragungen diskutiert. Eine Diskussion der technischen und theoretischen Aspekte der Arbeit ist bereits in den einzelnen Kapiteln enthalten. Abschließend wird in 9.3 ein Ausblick auf weitere Forschungsperspektiven gegeben, die durch die im Rahmen dieser Arbeit gewonnenen Ergebnisse eröffnet werden.

## 9.1 Zusammenfassung

Die Arbeit hat die Eignung hypermedialer Navigation zur Nutzung in Vorlesungsaufzeichnungen unter Berücksichtigung medienbedingter und umfeldbedingter Faktoren gezeigt.
Dazu wurden zunächst Rahmenbedingungen des Einsatzes und der Produktion von Vorlesungsaufzeichnungen analysiert. Darauf aufbauend wurden Anforderungen and die Konzeption und Umsetzung von Navigationskonzepten für Vorlesungsaufzeichnungen erarbeitet. Hierbei wurde zunächst die theoretische Eignung von hypermedialer Navigation für Vorlesungsaufzeichnungen gezeigt. Eine Betrachtung existierender Systeme zur Vorlesungsaufzeichnung hat ergeben, dass einzelne hypermediale Navigationselemente bereits in vielen Systemen vorhanden sind, jedoch bis auf das im Rahmen dieser Arbeit entwickelte hypermediale Navigationskonzept kein weiteres bekannt ist. Die Umsetzbarkeit des Navigationskonzeptes wurde durch eine konkrete Implementation gezeigt. Die zur Nutzung des Navigationskonzeptes nötigen Aufbereitungs- und Indizierungsschritte werden in der implementierten Produktionskette vollständig automatisiert durchgeführt. Das im Rahmen der Arbeit entwickelte System zur Vorlesungsaufzeichnung erfüllt damit nicht nur die in Kapitel 4 erarbeiteten Anforderungen an ein Navigationskonzept für Vorlesungsaufzeichnungen sondern auch die in Kapitel 1 und 2 dargestellten Bedingungen für den Einsatz im Regelbetrieb von Hochschulen.

## 9.2 Diskussion

Seit Beginn der Entwicklung des virtPresenter im Sommersemester 2003 wurde das System in verschiedenen Entwicklungsstufen an den Universitäten Osnabrück und Oldenburg sowie an der Fachhochschule Osnabrück eingesetzt. Im Rahmen dieser Einsätze wurde eine Vielzahl von Einsatzerfahrungen gesammelt. In einigen der aufgezeichneten Lehrveranstaltungen wurden von der Gruppe Usablity und Evaluation des Zentrum virtUOS der Universität Osnabrück Nutzerbefragungen zum Einsatz des Systems durchgeführt. Darüber hinaus begleitete die Gruppe Usablity und Evaluation die Entwicklung mit einer heuristische Evaluation einer Prototyp-Version des Interfaces und einem Benutzertest. Im weiteren Verlauf dieses Abschnittes werden die gesammelten Einsatzerfahrungen und die Ergebnisse der Nutzerbefragungen sowie der heuristischen Evaluation diskutiert.

### 9.2.1 Einsatzerfahrungen

In der Anfangsphase der Entwicklung des Systems wurden viele Produktionsschritte zur Erstellung der webfähigen Aufzeichnung manuell durchgeführt. Da die Entwicklung parallel zum Einsatz in mehreren Lehrveranstaltungen stattfand, bedeutete dies einen enormen Arbeitsaufwand zum Schnitt und zur Konvertierung von Videoaufzeichnungen und zur Konvertierung von PowerPoint in SVG. Die in 1.1.1 geschilderten Schätzungen zum Aufwand von mehreren Stunden zur Erstellung von Inhalten für eine Stunde Lernzeit können in diesem Kontext bestätigt werden. Die in 8.1 beschriebene Zerlegung von PowerPoint-Folien in Einzelschritte wurde in der ersten Version des Systems durch ein PowerPoint-Makro in

Visual Basic realisiert. Das Makro implementiere nur Teile der in 8.1 dargestellten Funktionalität, war aber aufgrund der Implementation in Visual Basic vergleichsweise unübersichtlich. Da das Makro keine Prozessüberwachung der in 7.3.1.6 beschriebenen Form enthielt, wurde die Zerlegung von PowerPoint-Foliensätzen häufig von Abstürzen der Applikation PowerPoint begleitet. Zur Aufzeichnung der Videoaufnahme wurden zwei verschiedene Strategien eingesetzt, zum einen die direkte Aufzeichnung über den Real-Producer und zum anderen eine nachträgliche Digitalisierung einer Aufnahme mit einer Videokamera mit dem Real-Producer. Ein Nachteil der direkten Aufzeichnung ist die Tatsache, dass der Real-Producer sich bei Aufzeichnungen mit höherer Qualität nicht immer als stabil erwiesen hat. Hauptnachteil der nachträglichen Digitalisierung ist der erhöhte organisatorische Aufwand und die mit der Digitalisierung verbundene Wartezeit zwischen Arbeitsschritten.
Durch die in Kapitel 1 beschriebene Automatisierung der Arbeitsschritte zur Produktion von Vorlesungsaufzeichnungen konnten diese manuellen Arbeitsschritte eliminiert werden. Auch konnte der Zeitraum zwischen Aufzeichnung und Bereitstellung der Mitschnitte minimiert werden, da in der automatisierten Produktionskette keine Wartezeiten zwischen einzelnen Prozessschritten anfallen. Beim Einsatz der Prozesskette in verschiedenen Prototypversionen haben sich jedoch auch Fehlerquellen gezeigt, die vom technisch bedingten Ausfall einiger Rechner über Stromausfälle und das Erreichen der Festplattenkapazität auf den Rechnern zur Videoaufzeichnung und –Konvertierung bis hin zu Defekten in der Netzwerktechnik von Hörsälen reichen. Vielen dieser Fehlerquellen wird in der aktuellen Implementation der Produktionskette begegnet. Beispiele hierfür sind die Serialisierung der Auftragsverwaltungsqueues im Videokonverter- und im Generatordienst (vgl. 7.3.2.1 und 7.3.2.2.1) und die Integration einer Speicherplatzmeldung in das Kommunikationsprotokoll zwischen Aufzeichnungskomponente und Videoaufzeichnungsrechner. Auf die automatisierte Löschung von Dateien in Ursprungsformaten wie PowerPoint oder MPEG-2 wurde in der eingesetzten Version der Prozesskette bewusst verzichtet, um die Möglichkeit zur Archivierung der Ursprungsdateien beispielsweise zur Konvertierung in andere Endformate zu erhalten.

### 9.2.2 Nutzerbefragungen

Im Jahr 2006 wurden in drei verschiedenen Veranstaltungen, in denen der virtPresenter eingesetzt wurde, Nutzerbefragungen zum Umgang mit dem System durchgeführt. Es handelte sich dabei jeweils um eine Veranstaltung aus dem Bereich Geisteswissenschaften, Wirtschaftswissenschaften und Informatik. In der Veranstaltung aus dem Bereich Informatik wurde ein standardisierter Fragebogen zur Evaluation virtueller Lehrelemente eingesetzt, in den beiden anderen Veranstaltungen wurde ein speziell entwickelter Fragebogen verwendet.
Die geisteswissenschaftliche Veranstaltung wurde von zwei Dozentinnen gehalten, von denen nur eine aufgezeichnet wurde. Daher sind nur insgesamt vier Sitzungen der Veranstaltung aufgezeichnet worden. In der Veranstaltung wurde die Aufzeichnung zusätzlich zur Präsenzveranstaltung angeboten (vgl. das in 3.1 dargestellte Szenario). Die Aufzeichnung wurde von ca. der Hälfte der 27 Teilnehmer genutzt. Die Bedienbarkeit der Wiedergabesoftware wurde von den Nutzern im Durchschnitt als gut angesehen. Probleme hatten die Nutzer mit langen Lade- und Pufferungszeiten und der Installation der nötigen Plugins. Da der Großteil der Nutzer den Microsoft Internet Explorer nutzte und die Studierenden in der Veranstaltung darauf hingewiesen worden sind, dass das Wiedergabeinterface nur mit dem Internet Explorer einwandfrei funktioniert, stellte die Abhängigkeit des Interfaces vom Internet Explorer kein Problem dar.
In der wirtschaftswissenschaftlichen Veranstaltung wurden alle Sitzungen aufgezeichnet. Im Gegensatz zu der geisteswissenschaftlichen Veranstaltung handelte es sich um eine standortverteilte Lehrveranstaltung per Videokonferenz, zu der zusätzlich Mitschnitte angeboten worden. Das Szenario ist vergleichbar mit einer Kombination der in 3.2 und 3.3 dargestellten

Szenarien. Insgesamt nahmen 19 Studierende an der Veranstaltung teil. Alle Teilnehmer haben die Veranstaltungsmitschnitte genutzt. Im Gegensatz zu den Teilnehmern der geisteswissenschaftlichen Veranstaltung verwendete die Mehrzahl der Teilnehmer (12) dieser Veranstaltung den Mozilla Firefox zur Wiedergabe der Aufzeichnung. Dies führte in der Nutzerbefragung mehrfach zu dem Wunsch einer browserübergreifenden Lösung. Hinzu kam die Tatsache, dass einige der Teilnehmer den Hinweis, dass die Software nur im Internet Explorer funktionierte, nicht beachtet hatten und die Inkompatibilität mit dem Mozilla Firefox durch Installation verschiedener Plugins zu umgehen versuchten. Darüber hinaus wurde der Wunsch nach einer Version der Aufzeichnung zum Download geäußert.
Bei der Informatikveranstaltung handelte es sich wie bei der wirtschaftswissenschaftlichen Veranstaltung um eine standortverteilte Lehrveranstaltung per Videokonferenz mit einem zusätzlichen Angebot der Mitschnitte. An der Veranstaltung nahmen 24 Studierende teil, von denen 23 die Aufzeichnung genutzt haben. Welche Browser von den Studierenden eingesetzt wurden, wurde nicht erhoben, einige Nutzer äußerten jedoch den Wunsch nach einer Lösung, die auch im Mozilla Firefox und unter anderen Betriebsystemen als Windows lauffähig ist. Im Gegensatz zu den Studierenden der geisteswissenschaftlichen Veranstaltung bewerteten die Teilnehmer der Informatikveranstaltung die Lade- und Pufferungszeiten der Inhalte im Durchschnitt als eher akzeptabel.
Durch die Nutzerbefragungen haben sich damit vor allem Browserunabhängigkeit, eine Verbesserung der Lade- und Pufferungszeiten sowie die Möglichkeit zum Download der erstellten Mitschnitte als Verbesserungspotentiale des Systems gezeigt.

### 9.2.3 Heuristische Evaluationen und Benutzertest

Mit einer Vorversion des in Kapitel 1 dargestellten Interfaces wurde im Juni 2006 eine heuristische Evaluation mit 3 Evaluatoren an der Universität Osnabrück durchgeführt. Heuristische Evaluation ist eine qualitative Methode zur Bewertung der Usability von Benutzerschnittstellen, die zur Bewertung bereits implementierter Nutzerschnittstellen dient. Wie in [Nielsen 1992] gezeigt, sind 3-5 Evaluatoren für eine heuristische Evaluation ausreichend. Darüber hinaus wurde eine Prototyp-Version des Interfaces durch einen Benutzertest mit 9 Versuchspersonen evaluiert.
Eine Reihe von Detailergebnissen ist in das im Rahmen dieser Arbeit vorgestellte Benutzerinterface eingeflossen. So ist die Darstellung einiger Bedienelemente ein Ergebnis der Evaluation. Auch die Forderung nach einer plattformunabhängigen Lösung ist von einigen Nutzern geäußert worden, ließ sich jedoch aufgrund der verwendeten Datenformate RealVideo und SVG in der aktuellen Version des Systems nicht realisieren. Ein weiterer Kritikpunkt, der auch durch die verwendeten Medienformate bedingt ist, sind lange Ladezeiten bei der Initialisierung des Interfaces und bei der Pufferung von Videosequenzen.
Bezüglich des im Rahmen der Arbeit entwickelten Navigationskonzeptes hat sich gezeigt, dass einige Navigationshilfen wie maussensitive Folienbestandteile zur strukturbasierten Navigation oder die Markierung von Folien- und Animationsschritten in der Zeitleiste den Nutzern zum Großteil nicht auf Anhieb aufgefallen sind. Die Bedeutung der Footprint-Markierungen auf der Zeitleiste wurde nur von einem Drittel der Versuchspersonen ohne Hilfestellung erkannt. Die Funktionsweise des Backtracking hat sich bis auf einen Fall als unproblematisch gezeigt.
Als Ergebnis der Evaluation lässt sich damit festhalten, dass die Versuchpersonen das Interface zwar nutzen konnten, einige der Navigationsfunktionen jedoch erst nach Hinweisen durch den Versuchsleiter erkannt haben. In Anbetracht der Tatsache, dass es sich bei dem in Rahmen dieser Arbeit entwickelten Navigationskonzept um eine Anpassung hypermedialer Navigation an zeitbasierte Medien handelt, ist dieses Ergebnis allerdings nicht überraschend. Gerade vor dem Hintergrund der in der Einleitung zu Kapitel 5 erläuterten Transparenz

hypermedialer Navigation für Text- und Bildmedien kann bei der Umsetzung hypermedialer Navigation für zeitbasierte Medien nicht von einer intuitiven Bedienbarkeit aller Navigationsfunktionen beim ersten Nutzen des Interfaces ausgegangen werden. So sind beispielsweise Nutzer vor der Verbreitung des WWW Nutzer bei der Ansicht eines unterstrichenen und farblich hervorgehobenen Begriffes auf einem Computerbildschirm auch nicht auf den Gedanken gekommen, mit der Maus darauf zu klicken. Es ist daher umso wichtiger, dass die Evaluation gezeigt, hat, dass das Interface auch ohne Kenntnis aller Navigationsfunktionen verwendet werden kann. Damit sind Grundfunktionen des Interfaces intuitiv verwendbar, zur Nutzung des vollständigen Navigationskonzeptes ist jedoch eine Einführung oder längeres exploratives Arbeiten mit dem Interface nötig.

## 9.3 Ausblick

Wie die in 9.2 dargestellten Ergebnisse der Nutzerbfragungen und der heuristischen Evaluation zeigen, weist das in dieser Arbeit vorgestellte System einige Verbesserungspotentiale auf, die in Weiterentwicklungen berücksichtigt werden sollten. Außerdem stellt das in Kapitel 1 vorgestellte Navigationskonzept eine solide Grundlage für weiterführende Navigationsfunktionen dar, wie beispielhaft an der implementierten Erweiterung zur sozialen Navigation gezeigt wurde. Im Folgenden werden beide Punkte jeweils eingehender dargestellt.

### 9.3.1 Weiterentwicklungsmöglichkeiten des virtPresenter

Die in 9.2.2 und 9.2.3 erwähnten langen Lade- und Pufferungszeiten sind vor allem durch den Einsatz der Medienformate RealVideo und SVG in der aktuellen Version des virtPresenter bedingt. Ein weiterer Nachteil dieser Formate ist die Notwendigkeit der Installation von zwei verschiedenen Plugins zur Wiedergabe von jeweils SVG und RealVideo. Auch die Abhängigkeit des Wiedergabeinterfaces vom Internet Explorer liegt in der Kombination der beiden Medienformate begründet. Zu Beginn der Entwicklung des virtPresenter schienen SVG und RealVideo gut für die Realisierung von Vorlesungsaufzeichnungssystemen geeignet zu sein. So wurde RealVideo um 2004 in vielen Neuentwicklungen in diesem Bereich verwendet. SVG war eine Open-Source-Alternative zu Flash und wurde besonders von Adobe gefördert. Mit der Übernahme von Macromedia durch Adobe wurde Macromedia Flash zu Adobe Flash und Adobe stellte die Entwicklung von Plugins zur Wiedergabe von SVG ein. RealVideo wurde in den letzten Jahren nicht maßgeblich verbessert. Mittlerweile stehen beispielsweise mit Flashvideo Alternativen zur Verfügung, die deutlich kürzere Ladezeiten bei navigationsbedingten Sprüngen im Video bieten. Eine Re-Implementation des Wiedergabeinterfaces unter Verwendung anderer Medienformate wäre daher sinnvoll. In [Mertens, Ketterl & Vornberger 2006] werden die Möglichkeiten skizziert, die sich hier durch die Verwendung von Flash zur Foliendarstellung und Flash-Video zur Wiedergabe des Dozentenvideos ergeben. Durch das Flex-Framework können die beiden Formate in einer einheitlichen Oberfläche dargestellt werden. Ein weiterer Vorteil wäre bei einer derartigen Lösung dadurch gegeben, dass Wiedergabekomponenten für Flash deutlich weiter verbreitet sind als für SVG und dass beide Fromate mit ein und demselben Plugin wiedergegeben werden können.

### 9.3.2 Weiterführende Forschungsperspektiven

Durch die Ergebnisse der vorliegenden Arbeit werden in mehreren Bereichen weiterführende Forschungsperspektiven eröffnet. Ein Beispiel ist die Umsetzung fortschrittlicher Navigationsansätze, die auf hypermedialer Navigation aufbauen (vgl. 6.3). Besonders hervorzuheben sind hier Erweiterungsmöglichkeiten der bereits implementierten sozialen Navigationskom-

ponente. Hier ist beispielsweise die Integration gruppenbasierter Filterung und Anzeige von Nutzerdaten denkbar.

Auch können die im Rahmen der Arbeit gewonnenen Erkenntnisse zur Verbindung von zeit- und strukturbasierter Navigation als Grundlage für weitere Forschungen dienen. Eine Zusammenfassung der wesentlichen Ergebnisse der vorliegenden Arbeit im Hinblick auf die Verbindung von zeit- und strukturbasierter Navigation in Vorlesungsaufzeichnungen ist in [Mertens, Brusilovsky, Ishchenko & Vornberger 2006] zu finden. Mögliche Weiterentwicklungen des Interfaces könnten beispielsweise an Fisheye-Views [Furnas 1986] angelehnt werden und den jeweils aktuellen Ausschnitt der Zeitachse vergrößert darstellen. Durch eine zusätzliche Verzerrung der Zeitachse in diesem Ausschnitt könnte der betreffende Teil der Zeitachse so transformiert werden, dass Foliengrenzen auf der Zeitachse genau mit den Grenzen der jeweiligen Folie im Überblick übereinstimmen. Zusätzlich würden derartig modifizierte Fisheye-Views zeitbasierte Navigation in einer höheren Auflösung ermöglichen.

Darüber hinaus bieten sich durch die im Rahmen dieser Arbeit entwickelte Produktionskette Möglichkeiten zur Integration weiterer Ausgabemedien wie beispielsweise Podcasts in die vollautomatische Produktion von Vorlesungsaufzeichnungen.

# A. Verzeichnis der Webverweise


[1] Webseiten der Vorlesung Algorithmentheorie im Rahmen des Universitären Lehrverbundes Informatik (Zugriff: 26.02.2007)
http://ad.informatik.uni-freiburg.de/lehre/ws0203/algorithmentheorie/download/at_lectures/at62.htm

[2] Authoring on the Fly (Zugriff: 26.02.2007)
http://ad.informatik.uni-freiburg.de/mmgroup/aof/index.html.de

[3] DFG-Verbundprojekt V3D2: Erzeugen multimedialer Dokumente on the Fly (Zugriff: 26.02.2007)
http://ad.informatik.uni-freiburg.de/mmgroup/Projects/V3D2/project.html

[4] The Apache ANT Project (Zugriff: 26.02.2007)
http://ant.apache.org

[5] ePresence Community (Zugriff: 26.02.2007)
http://blog.epresence.kmdi.utoronto.ca

[6] Linux-Kompendium: Netzwerkzeitprotokoll (Zugriff: 26.02.2007)
http://de.wikibooks.org/wiki/Linux-Kompendium%3A_Netzwerkzeitprotokoll

[7] Wikipedia: Document Object Model (Zugriff: 26.02.2007)
http://de.wikipedia.org/wiki/Document_Object_Model

[8] Wikipedia: Liste der SVG-Unterstützung in Browsern (Zugriff: 26.02.2007)
http://de.wikipedia.org/wiki/Liste_der_SVG-Unterst%C3%BCtzung_in_Browsern

[9] eClass Project: Lecture (Zugriff 20.12.2004. mittlerweile offline)
http://eclass.cc.gatech.edu/zenpad/db/showSession.php3?courseID=116&courseDir=cs7470_01_Spring&lectureID=2024&lectureDir=Jan.17.2001.1&LeNote=OFF&CoWeb=OFF&HWriting=OFF&Mode=ALL&Style=NOMEDIA&extSlide=

[10] eTEACH (Zugriff: 26.02.2007)
http://eteach.engr.wisc.edu/newEteach/home.html

[11] Microsoft Corporation: Installer Tool (Installutil.exe) (Zugriff: 26.02.2007)
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cptools/html/cpconinstallerutilityinstallutilexe.asp

[12] Microsoft Corporation: Microsoft Office Development with Visual Studio (Zugriff: 26.02.2007)
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnoxpta/html/vsofficedev.asp

[13] Microsoft Corporation: PowerPoint 2003 VBA Language Reference (Zugriff: 26.02.2007)
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbapp11/html/ppproSchemeColor1_HV05193605.asp

[14] Microsoft Corporation: Windows Driver Kit: Print Devices Choosing User Mode or Kernel Mode (Zugriff: 26.02.2007)
http://msdn2.microsoft.com/en-us/library/ms801074.aspx

[15] Microsoft Corporation: System.Diagnostics.Process FAQ (Zugriff: 26.02.2007)
http://msdn2.microsoft.com/en-us/netframework/aa569609.aspx#Question7

[16] RIPPLES: Research Projects (Zugriff: 26.02.2007)
http://ripples.cs.umass.edu/research.php#manic2.0

[17] RealNetworks: Embedded RealPlayer Extended Functionality Guide (Zugriff: 26.02.2007)
http://service.real.com/help/library/guides/extend/embed.htm

[18] RealNetworks: Helix Producer User's (Zugriff: 26.02.2007)
http://service.real.com/help/library/guides/helixproducer/Producer.htm

[19] RealNetworks: RealOne Player Scripting Guide (Zugriff: 26.02.2007) http://service.real.com/help/library/guides/realone/ScriptingGuide/HTML/realscript.htm

[20] Microsoft Corporation: Knowledge-Base Artikel Nr. 308825 (Zugriff: 26.02.2007) http://support.microsoft.com/default.aspx?scid=kb;en-us;308825

[21] Microsoft Corporation: Knowledge-Base Artikel Nr. 254009 (Zugriff: 26.02.2007) http://support.microsoft.com/kb/254009/EN-US/

[22] Microsoft Corporation: Knowledge-Base Artikel Nr. 257757 (Zugriff: 26.02.2007) http://support.microsoft.com/kb/257757/en-us

[23] Microsoft Corporation: Knowledge-Base Artikel Nr. 311452 (Zugriff: 26.02.2007) http://support.microsoft.com/kb/311452/en-us

[24] Microsoft Corporation: Knowledge-Base Artikel Nr. 827405 (Zugriff: 26.02.2007) http://support.microsoft.com/kb/827405/en-us

[25] TeleTeachingTool (Zugriff: 26.02.2007) http://teleteaching.uni-trier.de/

[26] Adobe Systems Incorporated: Adobe Illustrator (Zugriff: 26.02.2007) http://www.adobe.com/products/illustrator/

[27] Adobe Systems Incorporated: Macromedia Flash and Shockwave Players (Zugriff: 26.02.2007) http://www.adobe.com/products/player_census/npd/

[28] Adobe Systems Incorporated: Download - Adobe Universal PostScript Windows Driver Installer 1.0.6 (Zugriff: 26.02.2007) http://www.adobe.com/support/downloads/detail.jsp?ftpID=1502

[29] Adobe Systems Incorporated: Adobe to Discontinue Adobe SVG Viewer (Zugriff: 26.02.2007) http://www.adobe.com/svg/eol.html

[30] eClass (Zugriff: 26.02.2007) http://www.cc.gatech.edu/fce/eclass

[31] ePresence Interactive Media (Zugriff: 26.02.2007) http://www.epresence.tv/

[32] ePresence Archive (Zugriff: 26.02.2007) http://www.epresence.tv/mediacontent/website_archived.aspx

[33] ic#code: Download Our Most Current Build (Zugriff: 26.02.2007) http://www.icsharpcode.net/opensource/sharpziplib/Download.aspx

[34] Network Working Group Request for Comments: 959, File Transfer Protocol (FTP) (Zugriff: 26.02.2007) http://www.ietf.org/rfc/rfc959.txt

[35] Inkscape (Zugriff 26.02.2007) http://www.inkscape.org/

[36] Isoph: Articulate Presenter (Zugriff 26.02.2007) http://www.isoph.com/presenter.htm

[37] K-Lite Codec Pack (Zugriff 26.02.2007) http://www.k-litecodecpack.com

[38] Mike Koss: Object Oriented Programming in JavaScript (Zugriff 26.02.2007) http://www.mckoss.com/jscript/object.htm

[39] Microsoft Corporation: Office XP PIAs (Zugriff 26.02.2007) http://www.microsoft.com/downloads/details.aspx?FamilyId=C41BD61E-3060-4F71-A6B4-01FEBA508E52&displaylang=en

[40] RealNetworks: RealProducer (Zugriff 26.02.2007) http://www.realnetworks.com/products/producer/index.html

[41] Software Mechanics Pty Ltd: SVGmaker (Zugriff 26.02.2007)
http://www.svgmaker.com/

[42] Universitärer Lehrverbund Informatik (Zugriff 26.02.2007)
http://www.uli-campus.de/

[43] Virtuelle Hochschule Oberrhein (Zugriff 26.02.2007)
http://www.viror.de/

[44] W3C: Mathematical Markup Language (MathML™) 1.01 Specification (Zugriff 26.02.2007)
http://www.w3.org/1999/07/REC-MathML-19990707

[45] W3C: Synchronized Multimedia (Zugriff 26.02.2007)
http://www.w3.org/AudioVideo/#SMIL

[46] W3C: Scalable Vector Graphics (SVG) (Zugriff 26.02.2007)
http://www.w3.org/Graphics/SVG/

[47] W3C: Scalable Vector Graphics (SVG) 1.1 Specification (Zugriff 26.02.2007)
http://www.w3.org/TR/SVG/

# B. Literaturverzeichnis

[Abowd 1999] Abowd, G. D.: **Classroom 2000: An experiment with instrumenttation of a living education environment.** In *IBM Systems Journal*, 38(4). S. 508-530. 1999.

[Abowd, Graca Pimentel, Ishiguro, Kerimbaev & Guzdial 1999] Abowd, G. D., Grace Pimentel, M., Ishiguro, Y., Kerimbaev, B. & Guzdial, M.: **Anchoring discussions in lecture: An approach to collaboratively extending classroom digital media.** In *Proceedings of Computer Supported Collaborative Learning (CSCL'99)*. December, 1999. S. 11-19.

[Adcock 2004] Adcock, V. T.: **Implementing an integrated SVG application for real time dynamically generated Internet mapping**, *SVG Open 2004*, Tokyo, Japan. Sept 7-10, 2004. http://www.svgopen.org/2004/papers/SVG_Open_Abstract/

[Adobe Systems Incorporated 1999] **PostScript® Language Reference**. 3. Auflage. Addison-Wesley. 1999.

[Adrion, Burleson, Cooper, Israel, Kurose & Watts 2003] Adrion, W. R., Burleson, W., Cooper, W., Israel, W. L., Kurose, J & Watts, K.: **EXCITE: Enabling X-- Campus Information Technology Education**. In *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference*. Boulder, Colorado, November 2003. Band: 3, S4F S. 18-23.

[Albahari, Drayton & Merrill 2001] Albahari, B., Drayton, P. & Merrill, B.: **C# Essentials**. 2. Auflage 2001. O'Reilly.

[Almeida 2003] Almeida, J.: **Streaming Content Distribution Networks with Minimum Delivery Cost**. Ph.D. Thesis. University of Wisconsin - Madison. 2003.

[Altman, Chen & Low 2002] Altman, E., Chen, Y & Low, W. C.: **Semantic exploration of lecture videos**. In *Proceedings of ACM Multimedia 2002*. S. 416–417.

[Anderson, Hoyer, Wolfman & Anderson 2004] Anderson, R. J., Hoyer, C., Wolfman, S. A. & Anderson, R.: **A study of digital ink in lecture presentation**. In *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems*. 24.-29. April 2004, Wien, Österreich. S.567-574.

[AOFedit 2004] **AOF-Editor - Eine kleine Einführung**. Albert-Ludwigs-Universität Freiburg. 2004. http://ad.informatik.uni-freiburg.de/mmgroup/aof/Documentation/Download/AOF-Editor.pdf

[AOFwb 2000 ] **AOF Whiteboard AOFwb - Benutzerhandbuch** (Version 1.4.6), Albert-Ludwigs-Universität Freiburg. September 2000. http://ad.informatik.uni-freiburg.de/mmgroup/aof/Documentation/Download/aofwb_handbuch.pdf

[Arons 1993] Arons, B.: **SpeechSkimmer: Interactively Skimming Recorded Speech**. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. November 1993. S. 187-196.

[Aula, Jhaveri & Käki 2005] Aula, A., Jhaveri, N. & Käki, M.: **Information Search and Re-access Strategies of Experienced Web Users**. *In Proceedings of the 14th international conference on World Wide Web*. Chiba, Japan. 2005. S. 583 – 592.

[Bacher & Müller 1998] Bacher, C. & Müller, R.: **Generalized Replay of Multi-Streamed Authored Documents**. In *Proceedings of ED-MEDIA 1998*. Freiburg im Breisgau. Juni 1998.

[Bacher & Ottmann 1995] Bacher, C. & Ottmann, T.: **Authoring on the Fly**. In *Journal of Universal Computer Science*, Vol. 1, No. 10, Oktober 1995.

[Bacher 1999] Bacher, C.: **Das Authoring on the Fly System**. Dissertation, Albert-Ludwigs-Universität Freiburg, 1999.

[Bacher, Müller, Ottmann & Will 1997] Bacher, C., Müller, R., Ottmann, T. & Will, M.: **Authoring on the Fly. A new way of integrating telepresentations and courseware production**. In *Proceedings of ICCE 97*, Kuching, Sarawak, Malaysia, Dezember 1997.

[Baecker 2002a] Baecker, R. M.: **Showing Instead of Telling**. In *Proceedings of ACM SIGDOC 2002*. S. 10-16.

[Baecker 2002b] Baecker, R. M.: **Highly interactive webcasting with structured archives**. *ACM CSCW Conference Supplement*. 2002. S. 175-176.

[Baecker 2003] Baecker, R. M.: **A principled design for scalable internet visual communications with rich media, interactivity, and structured archives**. In: *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research*. IBM Centre for Advanced Studies Conference, Toronto, Ontario, Canada. 2003. S. 16 – 29.

[Baecker, Moore & Zijdemans 2003] Baecker, R. M., Moore, G. & Zijdemans, A.: **Reinventing the Lecture: Webcasting made Interactive**. In *Proceedings of HCI International 2003*. Juni 2003. Lawrence Erlbaum Associates, Volume 1. S. 896-900.

[Baecker, Wolf & Rankin 2004] Baecker, R. M., Wolf, P. & Rankin, K.: **The ePresence Interactive Webcasting System: Technology Overview and Current Research Issues**. In *Proceedings of Elearn 2004*. Washington, DC, USA. S. 2532-2537.

[Baez-Franceschi, Dinshaw, Evans & Van Nieuwenhuise 2006] Baez-Franceschi, S., Dinshaw, A., Evans, I. & Van Nieuwenhuise, D.: **Drag and Drop Streaming: an Inexpensive Method for Recording and Delivering Lectures is Becoming the Next Revolution in E-Learning**. In *Proceedings of E-Learn 2006*. Honolulu, HI, USA, 13-17 Oktober 2006. S. 2715-2720.

[Barrett 2004] Barrett, D. J.: **The Power of PowerPoint: Providing MBAs a Leadership Edge**. In: *Exploring the Impact of Technology on Business Communication Practice and Pedagogy - Proceedings from the 69th AnnualConvention of The Association for Business Communication*. 25.-29. Oktober 2004, Cambridge, Massachusetts. S. 127-152.

[Belenkaia, Lienhard, Mohamed & Ottmann 2004] Belenkaia, L., Lienhard, J., Mohamed, K. A. & Ottmann, T.: **Creation, Presentation, Capture, and Replay of Freehand Writing in e-Lecture Scenarios**. Technischer Bericht, Albert-Ludwigs-Universität Freiburg,

http://ad.informatik.uni-freiburg.de/mmgroup/Projects/f-moll/fmoll-Dateien/bericht.pdf

[Bell, Cockburn, McKenzie & Vargo 2001] Bell, T., Cockburn, A., McKenzie, B. & Vargo, J.: **Flexible Delivery Damaging to Learning? Lessons from the Canterbury Digital Lectures Project**. In: *Proceedings of ED-MEDIA 2001*. S. 117-122.

[Bernstein, Brown, Frisse, Glushko, Landow & Zellweger 1991] Bernstein, M., Brown, P. J., Frisse, M. E., Glushko, R. J., Landow, G. P. & Zellweger, P.: **Structure, Navigation, and Hypertext: The Status of the Navigation Problem**. In *Proceedings of ACM Hypertext 1991*. S. 363-366.

[Bianchi 2004] Bianchi, M.: **Automatic video production of lectures using an intelligent and aware environment**. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*. College Park, Maryland, 2004. ACM. S. 117 – 123.

[Bieber 2000] Bieber, M.: **Hypertext**. *Encyclopedia of Computer Science (4th Edition)*. Ralston, A., Reilly, E. & Hemmendinger, D. (Hrsg.)- Nature Publishing Group. 2000. S. 799-805.

[Bijnens, Bijnens & Vanbuel 2004] Bijnens, H., Bijnens, M. & Vanbuel, M.: **Streaming Media in the Classroom**. Education Highway. 2004.

[Breitner & Breuer 2006] Breitner, H. & Breuer, F.: **Mobile Aufzeichnung von Vorträgen Schnelle Installation – nachhaltiger Lernerfolg**. In *Technologie Informationen*. Ausgabe 3/2006, Herausgeber: Arbeitskreis der Technologietransferstellen niedersächsischer Hochschulen. Hannover. S. 5.

[Brotherton & Abowd 2004] Brotherton, J. A. & Abowd, G. D.: **Lessons Learned From eClass: Assessing Automated Capture and Access in the Classroom**. *ACM Transactions on Computer-Human Interaction*, Vol. 11, No. 2, Juni 2004. S. 121–155.

[Brotherton 2001] Brotherton, J. A.: **Enriching Everyday Experiences through the Automated Capture and Access of Live Experiences: eClass: Building, Observing and Understanding the Impact of Capture and Access in an Educational Domain**. Ph. D. Thesis. Georgia Tech, College of Computing. Dezember 2001.

[Brünken & Leutner 2001] Brünken R., & Leutner D.: **Aufmerksamkeitsverteilung oder Aufmerksamkeitsforschung? Empirische Ergebnisse zur "Split-Attention-Hypothese" beim Lernen mit Multimedia**. In *Unterrichtswissenschaften. Zeitschrift für Lernforschung*. Heft 4. 2001.

[Brusilovsky & Miller 2001] Brusilovsky, P. & Miller, P.: **Course Delivery Systems for the Virtual University**. In: Tschang F. T. & Della Senta, T. (Hrsg.): *Access to Knowledge: New Information Technologies and the Emergence of the Virtual University*. Amsterdam: Elsevier Science and International Association of Universities. 2001. S. 167-206.

[Brusilovsky 2000] Brusilovsky, P. (2000). **Web lectures: Electronic presentations in Web-based instruction**. *Syllabus Magazine* 13 (5). S. 18-23.

[Brusilovsky 2001] Brusilovsky, P.: **Adaptive Hypermedia**. In *User Modeling and User-Adapted Interaction* 11. Kluwer Academic Publishers. 2001. S. 87-110.

[Brusilovsky 2002] Brusilovsky, P.: **Web-based Interactive Visualization in an Information Retrieval Course**. In: Barker, P. & Rebelsky, S. (Hrsg.): *Proceedings of ED-MEDIA'2002 - World Conference on Educational Multimedia, Hypermedia and Telecommunications*. Denver, CO, USA. 24.-29. Juni 2002. AACE. S. 198-203.

[Brusilovsky 2004] Brusilovsky, P.: **Adaptive Navigation Support: From Adaptive Hypermedia to the Adaptive Web and Beyond**. In *PsychNology Journal*. Volume 2, Number 1. , 2004. S. 7 – 23.

[Brusilovsky, Chavan & Farzan 2004] Brusilovsky, P., Chavan, G., & Farzan, R.: **Social adaptive navigation support for open corpus electronic textbooks**. In: De Bra, P. & Nejdl, W. (Hrsg.): *Proceedings of Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004)*. Eindhoven, Niederlande. 23.-26. August 2004. S. 24-33.

[Bush 1945] Bush, V.: **As we may think**. *The Atlantic Monthly*. 176 (1). 1945. S: 101-108. `http://www.theatlantic.com/doc/194507/bush`

[Carmel, Crawford & Chen 1992] Carmel, E., Crawford, S., & Chen, H.: **Browsing in hypertext: A cognitive study**. *IEEE Transactions on System Man, and Cybernetics* 22 (5). S. 865-883.

[Casado & Torres 2004] Casado, R. & Torres, J. C.: **SVG Slide Presentation Editor**. In *SVG Open 2004*, Tokyo, Japan. Sept 7-10, 2004. `http://www.svgopen.org/2004/papers/SVGSlidePresentationEditor`

[Chambers & Specher 1980] Chambers, J. A., & Specher, J. W: **Computer Assisted Instruction: Current Trends and Critical Issues**. In *Communications of the ACM* 23 (6). S 332-342.

[Chan 2002] Chan, P.: **The Java(TM) Developers Almanac 1.4, Volume 1: Examples and Quick Reference (4th Edition)**. (Paperback). Prentice Hall PTR März 2002.

[Chan, Felber, Garofalakis & Rastogi 2001] Chan, C.-Y., Felber, P., Garofalakis, M. & Rastogi, R.: **Efficient filtering of XML documents with XPath expressions.** In *The VLDB Journal* (2002) 11. S. 354–379.

[Chen 2001] Chen, M.: **Design of a Virtual Auditorium**. In *Proceedings of ACM Multimedia 2001*. Ottawa, Canada, September 2001. S. 19-28.

[Chen, Ma, Meinel & Schillings 2003] Chen, T., Ma, M, Meinel, C. & Schillings, V.: **A novel approach to e-learning content creation**. In *Proceedings of the Second International Conference on Multimedia and Information & Communication Technologies in Education (m-ICTE)*, Badajoz, Spanien, Dezember 2003. S. 410–414.

[Cockburn, McKenzie & JasonSmith 2002] Cockburn, A., McKenzie, B. & JasonSmith, M.: **Pushing Back: Evaluating a New Behaviour for the Back and Forward Buttons in Web Browsers**. *International Journal of Human-Computer Studies*. 57(5). Academic Press. 2002. S. 397-414.

[Conklin 1987] Conklin, J.: **A survey of hypertext**. Technical Report STP-356~86. MCC. Austin, Texas. Dezember 1987. Revision 2.

[Cooperstock 2001] Cooperstock, J.R.: **The classroom of the future: enhancing education throug augmented reality**. In *Proceedings of HCI International 2001 Conference on Human-Computer Interaction*: New Orleans, USA. 2001. S. 688-692 .

[Cooperstock 2003] Cooperstock, J.R.: **Intelligent Classrooms need Intelligent Interfaces: How to Build a High-Tech Teaching Environment that Teachers can use?**. *American Society for Engineering Education*. 22.-25. Juni 2003, Nashville, Tennessee, USA. http://www.asee.org/acPapers/2003-1659_Final.pdf

[Cuthbert, Kubinec, Tanic, Ieong, Wei & Schlossberg 2005] Cuthbert, A, Kubinec, M., Tanis, D. O., Ieong, F., Wei, L. & Schlossberg, D.: **Advanced technology for streamlining the creation of ePortfolio resources and dynamically-indexing digital library assets: a case study from the digital chemistry project**. In *Proceedings of CHI 2005 extended abstracts on Human factors in computing systems*. 02.-07. April 2005, Portland, OR, USA. S. 972 - 987.

[Cox, Haskell, LeCun, Shahraray & Rabiner 1998] Cox, R. V., Haskell, B. G., LeCun, Y., Shahraray, B. & Rabiner, L. R.: **On the Applications of Multimedia Processing to Communications**. In *IEEE Proceedings* Vol. 86, No. 5. S. 755-824.

[Dannenberg & Capell 1997] Dannenberg, R.B., & Capell P.: **Are Just-In-Time Lectures Effective At Teaching?**. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. 1997. http://www.jitl.cs.cmu.edu/effectiv.pdf

[Day, Foley, Groeneweg & Van der Mast 2005] Day, J., Foley, J., Groeneweg, R. & Van der Mast, C.: **Enhancing the Classroom Learning Experience with Web Lectures**. In *Proceedings of the 2005 International Conference on Computers in education*. Asia-Pacific Society for Computers in Education. Singapur, Dezember 2005. S. 638-641.

[de Bruijn & Spence 2000] de Bruijn, O. & Spence, R: **Rapid Serial Visual Presentation: A space-time trade-off in information presentation**. In *Proceedings of Advanced Visual Interfaces, AVI’2000*. Palermo, Italien. S. 189-192.

[De Vries & Thampuran 2001] De Vries, K. & Thampuran, S.: **Challenges and Solutions to Creating and Delivering Multimedia Content for Asynchronous Courses**. In *Proceedings of IASTED International Conference on Computers and Advanced Technology in Education (CATE 2001)*. Banff, Canada. Juni 2001. http://manic.cs.umass.edu/about/pubs/challenges.pdf

[Deeths & Brunette 2001] Deeths, D. & Brunette, G.:**Using NTP to Control and Synchronize System Clocks - Part I: Introduction to NTP**. Sun Blueprints, 07/2001.

[Denoue, Hilbert, Adcock, Billsus & Cooper 2005] Denoue, L, Hilbert, D. M., Adcock, J, Billsus, D. & Cooper, M.: **ProjectorBox: Seamless presentation capture for classrooms**. In *Proceedings of E-Learn 2005*. Vancouver, Canada. S. 1986-1991.

[Dickson, Adrion & Hanson 2006] Dickson, P., Adrion, W. R. & Hanson, A.: **Automatic Capture of Significant Points in a Computer Based Presentation**. In *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'06)*. 11-13. Dezember 2006. San Diego, CA, USA. S. 921-926.

[Dieberger, Dourish, Höök, Resnick, Wexelblat 2000] Dieberger, A., Dourish, P., Höök, K., Resnick, P. & Wexelblat, A.: **Social Navigation: Techniques for Building More Usable Systems**. In *interactions*, 7, 6 (2000). S. 36-45.

[Dolzani & Ronchetti 2005] Dolzani, M. & Ronchetti, M.: **Lectures On DEmand: the architecture of a system for delivering traditional lectures over the Web**. In *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005*. S. 1702-1709.

[Dorai, Kermani & Stewart 2001] Dorai, C., Kermani, P. & Stewart, A.: **Elm-n: e-learning media navigator**. In *Proceedings of ACM Multimedia 2001*. S. 634–635.

[Dufour, Toms, Lewis & Baecker 2005] Dufour, C., Toms, E., Lewis, J. & Baecker, R.: **User strategies for handling information tasks in webcasts**. In *Proceedings ACM CHI 2005*, 2.-7. April 2005, Portland, OR. S. 1343-1346.

[Dunlop & Rijsbergen 1993] Dunlop, M. D. & van Rijsbergen, C.J.: **Hypermedia and Free Text Retrieval**. In *Information Processing and Management* 29 (3). S. 287-298.

[ECMA 1999] ECMA: **Standard ECMA-262 - ECMAScript Language Specification**.3te Auflage 1999
hppt://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf

[Effelsberg 2003] Effelsberg, W.: **Quo vadis alma mater? - Stand und Zukunft der virtuellen Lehre an unseren Hochschulen**. In: Stefan Pinkau, Thomas Gerke (Hrsg.). *E-learning. NMB-Projekte in den Ingenieurwissenschaften. Tagungsband zum Workshop der ingenieurwissenschaftlichen Projekte im bmb+f-Förderprogramm "Neue Medien in der Bildung"*. 25.-26. Juni 2003 an der Hochschule Anhalt in Dessau. S. 10-13.

[Effelsberg, Walke, Stüttgen & Ottmann 2003] Effelsberg, W., Walke, B., Stüttgen, H. & Ottmann, T.: **Abschlussbericht über das DFN-Projekt ANETTE – Applications and Network Technology for Teleteaching**. 2002.
http://webdoc.sub.gwdg.de/ebook/ah/dfn/anette.pdf

[Einhorn, Nejdl & Olbrich 2003] Einhorn, R., Nejdl, W., Olbrich, S.: **A Metadata Model for Capturing Presentations**. In *Proceedings of ICALT 2003 – 3rd IEEE International Conference on Advanced Learning Technologies*. 9.-11.Juli 2003. Athen, Griechenland. S. 110- 114.

[Eisenberg 2002] Eisenberg, J.: **SVG Essentials**. O'Reilly. 2002.

[Endicott 2000] Endicott, J.: **Growing up with PowerPoint**. In *Presentations Magazine*, Februar 2000. S. 61-66.

[Eskicioglu & Kopec 2003] Eskicioglu, A. M. & Kopec, D.: **The ideal multimedia-enabled classroom: Perspectives from psychology, education and information science**. In *Journal of Educational Multimedia and Hypermedia*, 12 (2), 2003. S. 199 – 221.

[Europäische Kommission 2002] Kommission der europäischen Gemeinschaften: **Vorschlag für einen Beschluss des europäischen Parlaments und des Rates über ein Mehrjahresprogramm (2004-2006) für die wirksame Integration von Informations- und Kommunikations-technologien (IKT) in die Systeme der allgemeinen und beruflichen Bildung in Europa (Programm „eLearning“)**. Brüssel 2002.
http://europa.eu.int/comm/education/programmes/elearning/doc/dec_de.pdf

[Farzan & Brusilovsky 2005] Farzan, R. & Brusilovsky, P.: **Social navigation support in E-Learning: What are real footprints**. In *Proceedings of IJCAI'05 Workshop on Intelligent Techniques for Web Personalization*. 1. August 2005. Edinburgh, U.K. S. 49-56.

[Fey 2002] Fey, A. (2002). **Audio vs. Video: Hilft Sehen beim Lernen?**. In *Unterrichtswissenschaften. Zeitschrift für Lernforschung*. 30 (2002) 4. S. 331-338.

[Flanagan 1997] Flanagan, D.: **JavaScript: The Definitive Guide**. 2te Auflage. O'Reilly. 1997.

[Foertsch, Moses, Strikwerda & Litzkow 2002] Foertsch, J., Moses, G., Strikwerda, J. & Litzkow, M.: **Reversing the lecture/homework paradigm using eTEACH web-based streaming video software**. In *Journal of Engineering Education*. 91 (3), 2002. S. 267-274.

[Friedl 1998] Friedl, J.: **Mastering Regular Expressions**. O'Reilly. 1998.

[Friedland 2004] G. Friedland: **Solving the Divided Attention Problem in Lecture Recordings**. In Knauer, C. (Grsg.): *2. Doktorandenworkshop des Instituts für Informatik*, November 2004. Freie Universität Berlin, Technischer Bericht B-04-15. S.. 24-28.

[Friedland 2006] G. Friedland: **Adaptive Audio and Video Processing for Electronic Chalkboard lectures**. Dissertation. Freie Universität Berlin. Oktober 2006.

[Furnas 1986] Furnas, G. W.: **Generalized fisheye views**. In *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*. Boston, Massachusetts, USA. 1986. S. 16 - 23.

[Gamma, Helm, Johnson & Vlissides 1995] Gamma, E., Helm, R., Johnson, R. E. & Vlissides, J: **Design Patterns. Elements of Reusable Object-Oriented Software: Elements of Reusable Object-Oriented Software**. Addison-Wesley. 1995.

[Girgensohn, Boreczky & Wilcox 2001] Girgensohn, A., Boreczky, J., & Wilcox, L.: **Keyframe-Based User Interfaces for Digital Video**. In *IEEE Computer*. Vol. 34(9).2001. S. 61-67.

[Gleicher, Heck & Wallick 2002] Gleicher, M.L., Heck, R. M., & Wallick, M. N.: **A framework for virtual Videography**. In *Proceedings of the 2nd international symposium on Smart graphics*.11-13 Juni 2002, Hawthorne, New York. S. 9-16.

[Glowalla 2004] Glowalla U.: **Utility und Usability von E-Learning am Beispiel von Lecture-on-demand Anwendungen**. In Steffens, C., Thüring, M. & Urbas, L. (Hrsg.): *Entwerfen und Gestalten. 5. Berliner Werkstatt Mensch-Maschine-Systeme. ZMMS Spektrum*. Band 18. Auszug aus Fortschritt-Berichte VDI, Reihe 22, Nr. 16, S. 603-621. Vdi Verlag. Düsseldorf.

[Gordon 2002] Gordon, A.: **The .NET and COM Interoperability Handbook**. Prentice Hall. Dezember 2002.

[Graca Pimentel, Abowd & Ishiguro 2000] Graca Pimentel, M., Abowd, G.,D. & Ishiguro, Y.: **Linking by interacting: a paradigm for authoring hypertext**. In *Proceedings of ACM Hypertext 2000*. S. 39-48.

[Graca Pimentel, Ishiguro, Abowd, Kerimbaev & Guzdial 2001] Graca Pimentel, M, Ishiguro, Y., Abowd, G. D., Kerimbaev, B. & Guzdial, M.: **Supporting Educational Activities through Dynamic Web Interfaces**. In *Interacting with Computers, special issue on interacting with the active Web*. Volume 13, Issue 3, Februar 2001. S. 3 53-374.

[Greenberg & Cockburn 1999] Greenberg, S. & Cockburn, A.: **Getting Back to Back: Alternate Behaviors for a Web Browser's Back Button**. In *Proceedings of the 5th Annual Human Factors and the Web Conference*. 3. Juni 1999, Gaithersburg, Maryland, USA. June 3th.
`http://www.itl.nist.gov/iaui/vvrg/hfweb/proceedings/greenberg/`

[Griffiths 2002] Griffiths, R. T.: **History of the Internet, Internet for Historians**. Universität Leiden, Niederlande. 2002.
`http://www.let.leidenuniv.nl/history/ivh/frame_theorie.html`

[Grossmann & Küchlin 2005] Grossmann, M. & Küchlin. W.: **MIN - Multimediale Module für Mathematik in Informatik und Naturwissenschaften**. In: J. M. Haake, U. Lucke und D. Tavangarian (Hrsg.): *Proceedings der DeLFI 2005: Die 3. Deutsche e-Learning Fachtagung Informatik*. Lectures Notes in Informatics (LNI), Gesellschaft für Informatik, September 2005. S. 201-212,

[Gulbransen, Bartlett, Bingham, Kachur, Rawlings & Watt 2002] Gulbransen, D., Bartlett, K., Bingham, E., Kachur, A., Rawlings, K. & Watt, A. H.: **Special Edition Using XML**. Que. 2002.

[Gütter, Neumann, Schill & Schreiter 1998] Gütter, D., Neumann, O., Schill, A. & Schreiter, G.: **Einsatz neuer Medien in der Rechnernetze-Ausbildung an der TU Dresden**. *GI-Fachtagung "Informatik und Ausbildung"*. Stuttgart, März 1998 . S. 186 – 197.

[Hall 2002] Hall, M.: **More Servlets and JavaServer Pages**. Sun Microsystems/Prentice Hall PTR. 2002.

[Hannemann, Thüring & Haake 1993] Hannemann, J., Thüring, M., & Haake, J. M.: **Hyperdocument Presentation: Facing the Interface**. *Arbeitspapiere der GMD 784*. GMD, St. Augustin, 1993.

[Hardman, Schmitz, van Ossenbruggen, ten Kate & Rutledge 2000] Hardman, L., Schmitz, P., van Ossenbruggen, J., ten Kate, W. & Rutledge , L.: **The Link vs. the Event: Activating and Deactivating Elements in Time-Based Hypermedia**. In *New Review of Hypermedia and Multimedia*, Volume 6, 2000. S. 89-109.

[Hartle, Bär, Trompler & Rößling 2005] Hartle, M., Bär, H., Trompler, C. & Rößling, G.: **Perspectives for Lecture Videos**. In *Proceedings of Euro-Par 2005 Parallel Processing: 11th International Euro-Par Conference*, Lissabon, Portugal, 30. August – 2. September 2005. S. 901-908.

[Hasselbring, W.; Appelrath, Sauer, Vornberger 2002] Hasselbring, W., Appelrath, H.-J., Sauer, O. & Vornberger, O.: **Verteiltes, virtuelles Softwareprojekt**. In: *Softwaretechnik-Trends*, Band 22 (3), 2002. S. 40-42.

[Haubold & Kender 2003] Haubold, A. & Kender, J. R.: **Analysis and Interface for Instructional Video**. *In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '03)*. 6.-8. Juli 2003, Baltimore, MD, USA. S. 705-708.

[He, Grudin & Gupta 2000] He, L., Grudin, J. & Gupta, A.: **Designing presentations for on-demand viewing**. In: *ACM 2000 Conference on Computer supported cooperative work*. 2000. S. 127-134.

[Heck, Wallick & Gleicher 2007] Heck, R., Wallick, M. & Gleicher, M.: **Virtual Videography**. In *ACM Transactions on Multimedia Computing, Communications, and Applications*. 3(1), Februar 2007.
http://doi.acm.org/10.1145/1198302.1198306

[Held, Ullrich, Neumann & Winter 2004] Held, G., Ullrich, T., Neumann, A. & Winter, A.: **Comparing .SWF (Shockwave Flash) and .SVG (Scalable Vector Graphics) file format specification**.
http://www.carto.net/papers/svg/comparison_flash_svg/

[Henkel 2004] Henkel, P.: **Quantitative Zugriffskontrolle in einer Laufzeitumgebung - Erweiterung der Common Language Infrastructure**. Diplomarbeit, Technische Universität Berlin, Januar 2004.

[Herzog, Thiesing & Vornberger 1999] Herzog, V., Thiesing, F. M. & Vornberger, O.: **Der Dokumentationsgenerator mas2tex**. *Workshop Neue Medien in Forschung und Lehre, GI-Jahrestagung 1999*, Paderborn.
http://www11.informatik.tu-muenchen.de/~brueggem/neueMedien99/vornberger1.pdf

[House & Pearlman 2003] House, L. & Pearlman, E.: **SVG for Web Developers**. Prentice Hall PTR. 2003.

[Huk, Steinke & Floto 2003] Huk, T., Steinke, M. & Floto, C.: **Helping teachers developing computer animations for improving learning in science education**. In: Crawford, C. et al. (Hrsg.), *Proceedings of the Society for Information Technology and Teacher Education International Conference 2003*. AACE. S. 3022-3025.

[Hurson & Kavi 1999] Hurson, A. R. & Kavi, K. M.: **Interactive teaching practices in small class sizes while cutting into the high cost of education**. In: *Proceedings of the International Conference on Engineering Education*.1999.
http://www.fs.vsb.cz/akce/1999/ICEE99/Proceedings

[Hürst 2001] Hürst, W.: **Browsing and navigation of AOF documents as an example for easy and comfortable interaction with general multimedia documents**. In *ECDL Workshop "Generalized Documents"* 2001, Darmstadt, September 2001.
http://ad.informatik.uni-freiburg.de/mitarbeiter/huerst/Publications/2001_ecdl-v3d2.pdf

[Hürst 2002] Hürst, W.: **Indexing, Searching, and Retrieving of Recorded Live Presentations with the AOF (Authoring on the Fly) Search Engine**. In *Proceedings of ACM SIGIR 2002*. Tampere, Finnland, August 2002. S. 447.

[Hürst 2003a] Hürst, W.: **Indexing, searching, and skimming of multimedia documents containing recorded lectures and live presentations**. In *Proceedings of the 11th Annual ACM International Conference on Multimedia*. Berkeley, CA, USA, Nov. 2003. ACM Press. S. 450-451.

[Hürst 2003b] Hürst, W.: **A Qualitative Study Towards Using Large Vocabulary Automatic Speech Recognition to Index Recorded Presentations for Search and**

**Access over the Web**. In *IADIS International Journal on WWW/Internet*. Volume I, Number 1, June 2003. S. 43-58.

[Hürst 2004] Wolfgang Hürst: **User Interfaces for Speech-Based Retrieval of Lecture Recordings**. In *Proceedings of ED-MEDIA 2004*. AACE, Lugano, Schweiz, Juni 2004. S. 4470-4477.

[Hürst 2005] Hürst, W.: **Multimediale Informationssuche in Vortrags- und Vorlesungsaufzeichnungen**. Dissertation. Albert-Ludwigs-Universität Freiburg im Breisgau. 2005.

[Hürst & Götz 2004] Hürst, W. & Götz, G: **Interface issues for interactive navigation and browsing of recorded lectures and presentations**. In Lorenzo Cantoni, L.& McLoughlin, C. (Hrsg): *Proceedings of the AACE World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2004)*. Lugano, SSchweitz. Juni 2004. AACE. S. 4464-4469.

[Hürst & Jarvers 2005] Hürst, W. & Jarvers, P.: **Interactive, Dynamic Video Browsing with the ZoomSlider Interface**. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2005)*. Amsterdam, Niederlande. S. 558 – 561.

[Hürst & Meyer 2004] Hürst, W. & Meyer, J.: **A New User Interface Design for Giving Lectures and Presentations**. In *Proceedings of E-Learn 2004, World Conference on E-Learning in Corporate Government, Healthcare & Higher Education.* Washington D.C., USA, November 2004. AACE. S. 2693-2700.

[Hürst & Stiegeler 2002] Hürst, W. & Stiegeler, R.: **User interfaces for browsing and navigation of continuous multimedia data**. In P*roceedings of the 2nd Nordic Conference on Human-Computer Interaction, NordiCHI 2002.* Aarhus, Dänemark. Oktober 2002. ACM Press. S. 267—270.

[Hürst, Lauer & Götz 2004] Hürst, W., Lauer, T. & Götz, G.: **Interactive manipulation of replay speed while listening to speech recordings**. In *Proceedings of the 12th ACM International Conference on Multimedia*. ACM Press, New York, NY, USA. S. 488-491.

[Hürst, Maass, Müller & Ottmann 2001] Hürst, W. Maass, G., Müller, R., & Ottmann. T.: **The `Authoring on the Fly' System for Automatic Presentation Recording**.In *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems, extended abstracts on Human factors in computing systems*. Seattle, WA, USA. 2001 S. 5-6.

[Hürst, Müller & Ottmann 2004] Hürst, W., Mueller, R. & Ottmann, T.: **The AOF Method for Production, Use, and Management of Instructional Media**. In *Proceedings of ICCE 2004, International Conference on Computers in Education*. Melbourne, Australien. Dezember 2004.

[Jantke 2005] Jantke, K.: **Innovative Technologien für das e-Learning: Für und Wider**. In *Proceedings of the Workshop on e-Learning 2005* HTWK Leipzig, 11.-12. Juli 2005. S. 9-30.

[Jarchow, Angilletta, Bönkost, Drawe, Ogurol, Salm & Wischnewsky 2002] Jarchow, S., Angilletta, S., Bönkost, K., Drawe, B., Ogurol, Y., Salm, M. & Wischnewsky, M.: **d-Lecture - der richtige Weg für freies, kollaboratives multimediales Lehren,**

**Lernen und Forschen? Eine Chance für die breite Weitergabe des Wissens?**. In Bernath, B. (Hrsg.): *Online Tutorien. Beiträge zum Spezialkongress „Distance Learning" der AG-F im Rahmen der LEARNTEC 2002*. Bibliotheks- und Informationssystem der Universität Oldenburg, S 181-190.

[Joukov & Chiueh 2003] Joukov, N. & Chiueh, T.: **LECTERN II: A Multimedia Lecture Capturing and Editing System**. In *Proceedings of. IEEE International Conference on Multimedia and Expo*, Baltimore MA, Juli 2003. Volume 2. S. 681-684.

[Jul & Furnas 1997] Jul, S., & Furnas, G. W.: **Navigation in Electronic Worlds**. In *SIGCHI Bulletin*, 29:4, Oktober 1997. S. 44-49.

[Kaasten, Greenberg & Edwards 2002] Kaasten, S., Greenberg, S. & Edwards, C.: **How People Recognize Previously Seen WWW Pages from Titles, URLs and Thumbnails**. In Faulkner, X., Finlay, J. &Detienne, F. (Hrsg.) *People and Computers XVI (Proceedings of Human Computer Interaction2002)*. BCS Conference Series. Springer Verlag. S. 247-265.

[Kaindl 2002] Kaindl, F.: **Streaming Media am Beispiel AOF**. Diplomarbeit. Universität Freiburg. 2002.
`http://ad.informatik.uni-freiburg.de/bibliothek/diplom/kaindl.pdf`

[Kandzia, Kraus & Ottmann 2004] Kandzia, P. T., Kraus, G. & Ottmann, T.: **Der Universitäre Lehrverbund Informatik - eine Bilanz**. In *GI Softwaretechnik-Trends*, Band 24 Heft 1, Februar 2004.

[Kelly & Goldsmith 2004] Kelly, S. D, & Goldsmith, L.: **Gesture and right hemisphere involvement in evaluating lecture material**. In *Gesture* 4:1 (2004). S. 25-42.

[Ketterl, Schmidt, Mertens & Morisse 2006] Ketterl, M, Schmidt, T., Mertens, R. & Morisse, K.: **Techniken und Einsatzszenarien für Podcasts in der universitären Lehre**. In *Workshopproceedings AudioLearning 2006 im Rahmen der 4. e-Learning Fachtagung der GI (DeLFI)*, September 2006. S. 81-90.

[Knaden & Rolf 2003] Knaden, A. & Rolf, R.: **Automatisierte Kamerasteuerung zur Übertragung von Seminaren und Vorlesungen**. In *e-Learning and beyond. Proceedings of the Workshop on e-Learning 2003*, HTWK Leipzig, 2003. S. 87-97.

[Knaden & Zettel 2002] Knaden, A. & Zettel, M.: **Das virtuelle Seminar – Liveübertragung standortübergreifender Lehrveranstaltungen via Internet**. Interner Bericht, Universität Osnabrück. 2002.

[Knight, Steinbach & White 2002] Knight, L. V., Steinbach, T. A. & White, J. D.: **An Alternative Approach to Web-based Education: Technology-intensive, Not Labor-Intensive**. In *Proceedings of ISECON 2002*. San Antonio.
`http://isedj.org/isecon/2002/223b/ISECON.2002.Knight.pdf`

[Knipping 2005] Knipping, L: **An Electronic Chalkboard for Classroom and Distance Teaching**. Dissertation. Freie Universität Berlin, Fachbereich für Mathematik und Informatik. Februar 2005.

[Krüger 2005] Krüger, M.: **Vortragsaufzeichnungen – Ein Querschnitt über die pädagogischen Forschungsergebnisse**. In Horz, H., Hürst, W., Ottmann, T., Rensing, C & Trahasch, S.: *eLectures- Einsatzmöglichkeiten, Herausforderungen und*

*Forschungsperspektiven*. In Lucke, U., Nölting, K., Tavangarian, D. (Hrsg.) *Workshop Proceedings DeLFI 2005 und GMW 2005*. Rostock, 13. – 16. September 2005. S. 31-36.

[Krüger, Heinrich & Jobmann 2004] Krüger, M., Heinrich, A., & Jobmann, K.: **Das iL2 - ein leicht zu bedienender Multimediaraum für eLearning**. In *DFN-Arbeitstagung über Kommunikationsnetze 2004*. S. 181-204.

[Krüger, Klie, Heinrich & Jobmann 2005] Krüger, M., Klie, T., Heinrich A. & Jobmann, K.: **Interdisziplinärer Erfahrungsbericht zum Lehren und Lernen mit dLectures.** In Breitner, M. & Hoppe, G. (Hrsg.): *in E-Learning - Einsatzkonzepte und Geschäftsmodelelle*. Physica-Verlag, August 2005. S. 53-72.

[Kühl 2002] Kühl, B.: **Objekt-Orientierung im Compilerbau**. Dissertation. Iniversität Osnabrück. 2002.

[Kumela, Watts & Adrion 2004] Kumela, D., Watts, K. & Adrion, W.R.: **Supporting constructivist learning in a multimedia presentation system**. In *Proceedings of Frontiers in Education, 2004. FIE 2004. 34th Annual Conference*. Volume 1. T1H. S. 18-23.

[Kunze 2006] Kunze, R.: **SVGWeather - Entwicklung einer SVG Web Mapping Applikation zur Visualisierung von vierdimensionalen Daten am Beispiel von Wettervorhersagedaten**. Dissertation. Universität Osnabrück. 2006.

[Kunze, Mertens & Vornberger 2005] Kunze, R., Mertens, R. & Vornberger, O.: **Dynamic and interactive visualization of weather data with SVG**. In *Proccedings of SVG open 2005, 4th Annual Conference on Scalable Vector Graphics*. Enschede, Niderlande. August 2005. http://www.svgopen.org/2005/papers/DynamicInteractiveWeatherData/index.html

[Lanham & Zhou 2004] Lanham, E. & Zhou, W.: **Giving lectures a voice for a cross-cultural audience**. In R. Atkinson, R., McBeath, C. Jonas-Dwyer D. & Phillips , R.(Hrsg), *Beyond the comfort zone: Proceedings of the 21st ASCILITE Conference*. 5.-8. Dezember 2004. Perth, Australien. http://www.ascilite.org.au/conferences/perth04/procs/lanham-poster.html

[Larsen, McIlhagga, Baert & Bundesen 2003] Larsen A., McIlhagga W., Baert J.& Bundesen C.: **Seeing or hearing? Perceptual independence, modality confusions, and crossmodal congruity effects with focused and divided attention**. In *Perception & Psychophysics*, Volume 65, Number 4, 1. May 2003. S. 568-574.

[Lauer & Busl 2007] Lauer, T. & Busl, S.: **Supporting Speech for Annotation and Asynchronous Discussion of Multimedia Presentations.** In *International Journal of Interactive Technology and Smart Education (ITSE)*, 4 (1). Februar 2007. Troubador publishing, UK. Erscheint demnächst.

[Lauer & Ottmann 2002] Lauer, T. & Ottmann, T.: **Means and Methods in Automatic Courseware Production: Experience and Technical Challenges**. In *Proccedings of the World Conference on E-Learning in Corp., Govt., Health., & Higher Education (E-Learn 2002).* S. 553-560.

[Lauer & Trahasch 2005] Lauer, T. & Trahasch, S.: **Begriffsbesprechung: Vorlesungsaufzeichnungen**. In *i-com Zeitschrift für interaktive und kooperative Medien*. 2005, 3. S. 61.

[Lea 1999] Lea, D.: **Concurrent Programming in Java**. Second Edition. Addison-Wesley, 1999.

[Lecolinet, Faure, Demeure, Moissinac & Pook 2005] Lecolinet, E., Faure, C., Demeure, I., Moissinac J.-C. & Pook, S.: **Augmentation de cours et de réunions dans un campus**. In *Proceedings of the 2nd French-speaking conference on Mobility and uibquity computing*. 31. Mai- 3. Juni 2005. Grenoble, France. ACM International Conference Proceeding Series, Vol. 120. S. 161-168.

[Leiner, Cerf, Clark, Kahn, Kleinrock, Lynch, Postel, Roberts & Wolff 2005] Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J., Roberts, L. G. & Wolff, S.: **A Brief History of the Internet**. Internet Society. 2005. `http://www.isoc.org/internet/history/brief.shtml`

[Li, Gupta, Sanocki, He & Rui 2000] Li, F. C., Gupta, A., Sanocki, E., He, L. & Rui Y.: **Browsing digital video**. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Den Haag, Niederlande. 2000. S. 169-176.

[Lienhard & Lauer 2002] Lienhard, J. & Lauer, T.: **Multi-layer recording as a new concept of combining lecture recording and students' handwritten notes**. In *Proceedings of ACM Multimedia 2002*. 1.-6. Dezember 2002. Juan-les-Pins, Frankreich. S. 335-338.

[Litzkow, Anderson, Bundy & Ewers 2004] Litzkow, M., Anderson, A., Bundy, C., B. & Ewers, N.: **Making Multi-media Web-based Lectures Accessible - Experiences, Problems, and Solutions**. In *online Proceedings of the 19th Annual International Conference on Technology and Persons with Disabilities*. 15-20 März 2004, Los Angeles CA, USA. `http://www.csun.edu/cod/conf/2004/proceedings/193.htm`

[Liu & Kender 2004] iu, T. & Kender, J. R.: **Lecture Videos for E-learning: Current Research and Challenges**. In *Proceedings of IEEE International Workshop on Multimedia Content-based Analysis and Retrieval*. 13.-15. Dezember 2004. Miami, Florida, USA. S. 574-578.

[Manber, Smith & Gopal 1997] Manber, U., Smith, M. & Gopal, B.: **Webglimpse: Combining browsing and searching**. In *Proceedings of the Usenix Technical Conference*. 1997. Los Angeles, CA. S.195-206.

[Masui, Kashiwagi & Borden 1995] Masui, T., Kashiwagi, K. & Borden, G.R.: **Elastic graphical interfaces for precise data manipulation**. In *Conference companion on Human factors in computing systems 1995*. ACM Press. S. 143-144.

[Mate, Velasquez & Sarmenta 2004] Mate, M. D., Velasquez, D. M. & Sarmenta, L. F. G.: **An Enhanced Lecture Viewer for eLearning**. In *Proceedings of the 3rd National Conference on eLearning*. Philippine eLearning Society, Ateneo de Manila University, August 2004. `http://discs.ateneo.net/lfgs/EVES-eLearning.pdf`

[Mayer 2001] Mayer R. E.: **Multimedia Learning**, Cambridge University Press, Cambridge UK. 2001.

[Meinel & Schillings 2002] Meinel, C. & Schillings, V.: **tele-TASK – Teleteaching Anywhere Solution Kit**. Forschungsbericht Nr. 2002-21 FB IV – Informatik, Universität Trier.

[Mertens, Brusilovsky, Ishchenko & Vornberger 2006] Mertens, R., Brusilovsky, P., Ishchenko, S. & Vornberger, O.: **Time and Structure Based Navigation in Web Lectures: Bridging a Dual Media Gap**. In *Proceedings of the World Conference on E-Learning, in Corporate, Government, Healthcare & Higher Education (E-Learn 2006).* Honolulu, HI, USA. 13.-17. Oktober 2006. S. 2929-2936.

[Mertens, Farzan & Brusilovsky 2006] Mertens, R., Farzan, R. & Brusilovsky, P.: **Social Navigation in Web Lectures**. In *Proceedings of ACM Hypertext 2006*. Odense, Dänemark, 23.-25. August 2006. S. 41-44.

[Mertens, Friedland & Krüger 2006] Mertens, R., Friedland, G. & Krüger, M.: **To See or Not To See: Layout Constraints, the Split Attention Problem and their Implications for the Design of Web Lecture Interfaces**. In *Proceedings of the World Conference on E-Learning, in Corporate, Government, Healthcare & Higher Education (E-Learn 2006)*. Honolulu, HI, USA, 13.-17. Oktober 2006. S. 2937-2943.

[Mertens, Ickerott, Witte & Vornberger 2005] Mertens, R., Ickerott, I.. Witte, T. & Vornberger, O.: **Entwicklung einer virtuellen Lernumgebung für eine Großveranstaltung im Grundstudium: Erfahrungen, Automatisierungspotentiale und Einschränkungen**. In *Proceedings of the Workshop on e-Learning 2005*. HTWK Leipzig, 11.-12. Juli 2005. S. 197-210.

[Mertens, Ketterl & Vornberger 2006] Mertens, R., Ketterl, M. & Vornberger, O.: **Interactive Content Overviews for Lecture Recordings**. In *Proceedings of IEEE International Symposium on Multimedia 2006*. Workshop on Multimedia Technologies for E-Learning (MTEL), San Diego, CA, USA, 11.-13. Dezember 2006. S. 933-937.

[Mertens, Ketterl & Vornberger 2007] Mertens, R., Ketterl, M. & Vornberger, O.: **The virtPresenter lecture recording system: Automated production of web lectures with interactive content overviews**. In *International Journal of Interactive Technology and Smart Education (ITSE),* 4 (1). Februar 2007. Troubador publishing, UK. S. 55-66.

[Mertens, Knaden, Krüger & Vornberger 2004] Mertens, R., Knaden, A., Krüger, A. & Vornberger, O.: **Einsatz von Vorlesungsaufzeichnungen im regulären Universitätsbetrieb.** In *Proceedings der 34. Jahrestagung der Gesellschaft für Informatik*. Workshop "Elektronische Unterstützung der Präsenzlehre". Ulm: 21 September 2004. S. 429-433.

[Mertens, Knaden, Thelen & Vornberger 2005a] Mertens, R., Knaden, A., Thelen, T. & Vornberger O.: **Kopplung von LMS und Vorlesungsaufzeichnungssystemen: Voraussetzungen und Potentiale**. In Horz, H., Hürst, W., Ottmann, T., Rensing, C & Trahasch, S.: *eLectures- Einsatzmöglichkeiten, Herausforderungen und Forschungsperspektiven*. In Lucke, U., Nölting, K., Tavangarian, D. (Hrsg.) *Workshop Proceedings DeLFI 2005 und GMW 2005*. Rostock, 13. - 16. September 2005. S. 43-48.

[Mertens, Knaden, Thelen & Vornberger 2005b] Mertens, R., Knaden, A., Thelen, T. & Vornberger, O.: **Perspektiven der Kopplung von LMS und Vorlesungsaufzeichnungenssystemen**. In i-*com Zeitschrift für interaktive und kooperative Medien.* Ausgabe 4, Dezember 2005, Oldenbourg Wissenschaftsverlag. S. 52-55.

[Mertens, Krüger & Vornberger 2004] Mertens, R., Krüger, A. & Vornberger, O**.: Einsatz von Vorlesungsaufzeichnungen**. In Hamborg, K.-C., Knaden, A. (Hrsg.): *Good Practice: netzbasiertes Lehren und Lernen an Universitäten - Erfahrungen mit verschiedenen Einsatzszenarien von e-Learning an der Universität Osnabrück.* epos Media, Osnabrück. S. 79-93.

[Miano 1999] Miano, J.: **Compressed Image File Formats: JPEG, PNG, GIF, Xbm, BMP**. 2. Auflage. ACM Press.1999.

[Miyahara, Kaiya & Kaijiri 2002] Miyahara, N., Kaiya H. & Kaijiri, K.: **A Web-Based VOD Clipping Tool for Efficient Private Review**. In *Proceedings of the 5th Joint Conference on Knowledge-Based Software Engineering (JCKBSE2002).* September 2002. S. 313-316.

[Moreno & Mayer 2000] Moreno R. & Mayer R.: **A Learner-Centred Approach to Multimedia Explanations: Deriving Instructional Design Princilpes from Cognitive Theory**. In *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*. 2(5), 2000. S. 1-7.

[Morkes & Nilsen 1997] Morkes, J. & Nielsen, J.: **Concise, SCANNABLE, and objective: How to write for the Web**.
`http://www.useit.com/papers/webwriting/writing.html`

[Moses, Litzkow, Foertsch & Strikwerda 2002] Moses, G., Litzkow, M., Foertsch, J. & Strikwerda, J.: **eTeach- A Proven Learning Technology for Education Reform**. In *Proceedings of IEEE – Frontiers in Education*. 6.-9. November 2002. Boston, MA, USA. S. 267-274.

[Mu 2005] Mu, X.: **Decoupling the Information Application from the Information Creation: Video as Learning Objects in Three-Tier Architecture**. In *Interdisciplinary Journal of Knowledge and Learning Objects*. 1, 2005. S. 109-125.

[Mühlhäuser & Trompler 2002] Mühlhäuser, M. & Trompler, C.: **Digital Lecture Halls Keep Teachers in the Mood and Learners in the Loop**. In *Proceedings of E-Learn 2002*. S. 714-721.

[Mukhopadhyay & Smith 1999] Mukhopadhyay, S. & Smith, B.: **Passive Capture and Structuring of Lectures**. In *Proceedings of ACM Multimedia ' 99.* Orlando, FL, USA. S. 477-487.

[Müller, Ottmann & Zhang 2002 ] Müller, R., Ottmann, T. & Zhang, H.: **Presentation Recording as a means to go Virtual for Campus-Based Universities**. In *Issues and Trends of IT Management in Contemporary Organizations: 2002 Information Resources Management Association International Conference*. Mai 2002, Seattle, WA, USA. S. 567-570.

[Munteanu, Baecker, Penn, Toms & James 2006] Munteanu, C., Baecker, R., Penn, G., Toms, E. G. & James, D: **The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives**. In *Proceedings of CHI 2006*. S. 493-502.

[Nielsen 1990a] Nielsen, J.: **Hypertext and Hypermedia**. Boston, MA. Academic Press. 1990.

[Nielsen 1990b] Nielsen, J.: **The Art of Navigating through Hypertext**. In *Communications of the ACM*. 33 (3). S. 297-310.

[Nielsen 1992] Nielsen, J.: **Finding usability problems through heuristic evaluation**. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Monterey, CA, USA. 1992. S. 373 – 380.

[Ollermann, Gruber & Hamborg 2003] Ollermann, F., Gruber, C. & Hamborg, C.-K.: **Bedeutung von Anforderungsanalysen für E-Learning-Szenarien**. In Dötsch, V., Schade, G. & Hering, K. (Hrsg.): *e-Learning and beyond. Proceedings of the Workshop on e-Learning.* 14.-15. Juli 2003, Leipzig: Hochschule für Technik, Wirtschaft und Kultur (FH), Fachbereich Informatik, Mathematik und Naturwissenschaften. S. 191-198.

[Olston & Chi 2003] Olston, C. & Chi, E. H.: **ScentTrails: Integrating Browsing and Searching on the Web**. In ACM Transaction on Computer-Human Interaction. Volume 10, Issue 3. September 2003. S. 177-197.

[Parmar 1993] Parmar, R.: **Hypertext for Writers: A Review of Software**. In *EJournal*. Volume 3 (3).

[Peterson 1999] Peterson, K. E.: **Force Your Way to the Foreground**. *VBPJ*. Februar 1999 S. 94- 97.
`http://www.ftponline.com/archives/premier/mgznarch/VBPJ/1999/02feb99/ap0299.pdf`

[Phillips & Kidson 2005] Phillips, D., Kidson, M.: **SVGmaker Content Server Guide for Integrators and Developers**. Version 2.0.6. Software Mechanics Pty Ltd.
`http://www.svgmaker.com/specs/SVGMCS_API.pdf`

[Ponceleon & Dieberger 2001] Ponceleon D., Dieberger A.: **Hierarchical Brushing in a Collection of Video Data**. In *Proceedings of HICSS'34 (Hawaii International Conference on Systems Science)*. Maui, HI, Januar 2001. S. 1654 -1661.

[Probets, Mong, Evans & Brailsford 2001] Probets, S. G., Mong, J., Evans, D. R. & Brailsford, D. F.: **Vector Graphics; from PostScript and Flash to SVG**. *In Proceedings of the ACM Symposium on Document Engineering (DocEng'01.)* Atlanta, Georgia. 9.–11. November 2001. S. 135–143.

[Rankin, Baecker & Wolf 2004] Baecker, R., Wolf, P. & Rankin, K.: **The ePresence Interactive Webcasting and Archiving System: Technology Overview and Current Research Issues**. In Richards, G. (Hrsg.): *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004*. Washingston, D.C. S. 2532-2537.

[Rechert 2003] Rechert, K.: **AOF2Web**. Studienarbeit durchgeführt am Lehrstuhl Algorithmen und Datenstrukturen - Multimedia-Forschungsgruppe, Albert-Ludwigs-Universität Freiburg, 2003.
`klaus.geekserver.net/Studienarbeit.ps`

[Robinson, Nagel, Glynn, Skinner, Watson & Evjen 2004] Robinson, S., Nagel, C., Glynn, J., Skinner, M. Watson, K. & Evjen, B: **Professional C#**. Third Edition. Wiley Publishing. 2004.

[Rohde &Thomas 2003] Rohde, P. & Thomas, W.: **Ein e-Lecture-System für die Theoretische Informatik**.In *DeLFI 2003, Tagungsband der 1. e-Learning Fachtagung Informatik*. 16.-18. September 2003. Garching bei München. LNI 37 GI 2003, S. 17-26.

[Rotard, Schweikhardt & Ertl 2002] Rotard, M., Schweikhardt, W. & Ertl, T.: **Verwendung von Scalable Vector Graphics und MathML in web-basierten Lernumgebungen**. In *Mensch und Computer 2002, Workshop CSCL: Neue Konzepte für die Gestaltung kooperativer Lernumgebungen*. Hamburg. 2002.
http://www.vis.uni-stuttgart.de/ger/research/pub/tr2002/TR2002-07-rotard.pdf

[Rowe, Harley, Pletcher & Lawrence 2001] Rowe, L. A., Harley, D., Pletcher, P. & Lawrence: **BIBS: A Lecture Webcasting System**. Center for Studies in Higher Education. Paper CSHE4-01. 20. März 2001.
http://repositories.cdlib.org/cshe/CSHE4-01

[Rubin & Bain 1999] Rubin, W. & Bain, M.: **Understanding DCOM**. Prentice Hall PTR. 1999.

[Samulat 2000] Samulat, P.: **Die genaue Zeit, bitte - Zeitsynchronisierung im Linux-Netzwerk**. *LINUX-MAGAZIN* 12/2000, S. 126-130.
http://www.linum.com/de/presse/linuxmagazin.htm

[Sand 2003] Sand, M.: **Design and Implementation of a System for Recording and Playback of Presentations**. Diplomarbeit. Universität des Saarlandes, Saarbrücken. 2003.

[Sanders 2001] Sanders, W.: **JavaScript Design**. New Riders. 2001.

[Schapira, De Vries & Pedregal-Martin 2001] Schapira, A., De Vries, K., & Pedregal-Martin, C.: **MANIC: An Open Source system to create and deliver courses over the internet**. In *Symposium on Applications and the Internet*. 2001. IEEE Computer Society Press. San Diego, CA. S. 21-26.

[Schertler & Bodendorf 2003] Schertler, M. & Bodendorf, F.: **Production Environment for Web-Based Video Lectures**. In: Rossett, A. (Hrsg.): *Proceedings of E-LEARN 2003*. AACE Publishing, Phoenix, AZ. 2003, S. 2401 - 2408.

[Shigeru & Masaaki 2002] Shigeru, S. & Masaaki, T.: **Geospatial Information Service System for Browser-phones utilizing PSVG**. In *Proceedings of SVG-Open 2002*. Zürich. Schweiz. 15.-17. Juli 2002.
http://www.svgopen.org/2002/papers/shimada_masaru_sitama__geospatial_infoservices_for_mobiles/

[Sim, Clarke, Holt & Cox 1999] Sim, S. E., Clarke, C. L. A. Holt, R. C. & Cox, A. M.: **Browsing and Searching Software Architectures**. In *Proceedings of the International Conference on Software Maintenance.* Oxford, England. September 1999. S. 381-390.

[Simmons 2004] Simmons,T.: **Does PowerPoint make you stupid?**. *Presentations*, 18(3), 2004. S. 25-31.

[Smeaton & Keogh 1999] Smeaton, A. F. & Keogh, G.: **An analysis of the use of virtual delivery of undergraduate lectures**. In *Computers and Education* 32, 1999. S. 83-94.

[Späth 2006] Späth, B.: **Das Aufzeichnen von Vorlesungen und die automatische Erstellung synchronisierter Online-Präsentationen**. Diplomarbeit. Bayerische Julius-Maximilians-Universität. Würzburg. August 2006.

[Sporer, Köstlbacher & Erbacher 2005] Sporer, T., Köstlbacher, A. & Erbacher, C.: **Projekt Knowledgebay - Erfahrungen mit der Produktion und Nutzung von Online-Vorlesungen durch Studierende**. In Lucke, U, Nölting, K. & Tavangarian, D. (Hrsg.): *Workshop Proceedings eLectures der DeLFI 2005 und GMW05*. S. 49-54.

[Stern, Steinberg, Lee, Padhye & Kurose 1997] Stern, M., Steinberg, H., Lee, H., Padhye, J. & MKurose, J.: **MANIC: Multimedia Asynchronous Networked Individualized Courseware**. In *Proceedings of World Conference on Educational Multimedia and Hypermedia*. Calgary, Kanada. 1997.

[Stockmann & Schäffer 2002] Stockmann, R. & Schäffer, E.: **Konzept zur Evaluation von E-Learning Angeboten im Rahmen von VISU (Virtuelle Saar-Universität)**. Centrum für Evaluation – Universität des Saarlandes. Saarbrücken 2002.

[Stringer 2003] Stringer K.: **Video technologies for Database course delivery**. In *Proceedings of Teaching, Learning and Assessment in Databases (TLAD)*. Coventry, England. 14. Juli 2003. S. 113-117.

[Subrahmanyam & Raghavan 2003] Subrahmanyam, D. V. V. R. & Raghavan, S. V.: **Architecture of e-Presence**. In *Proceedings of the 5th International Conference on Information Integration and Web-based Applications and Services (IIWAS 2003)*. Jakarta, Indonesien. September 15-17, 2003. S. 35-44.

[Tauscher 1996] Tauscher, L. M.: **Evaluating History Mechanisms: An Empirical Study of Reuse Patterns in World Wide Web Navigation**. M. Sc. Thesis. University of Calgary. Calgary, Alberta, Kanada. 1996.
`http://grouplab.cpsc.ucalgary.ca/papers/1996/96-Tauscher.Thesis/thesis/entire_thesis.pdf`

[Teague & Campbell 2003] Teague, J. & Campbell, M.: **SVG for Web Designers**. Wiley. 2003.

[Technical Advisory Service for Images 2006] Technical Advisory Service for Images: **File Formats for Teaching and Learning -Vector Images Part I: Overview**. Last reviewed: November 2006.
`http://www.tasi.ac.uk/advice/vector/pdf/tl_vector1.pdf`

[Theimer 2004] Theimer, F.: **Automatische Handschriftenerkennung in E-Kreide Dokumenten**. Technischer Bericht B 04-05. Freie Universität Berlin. 2004.

[Tolhurst 1995] Tolhurst, D.: **Hypertext, hypermedia, multimedia defined?**. In *Educational Technology*, 35(2), 1995. S. 21-26.

[Troelsen 2002] Troelsen, A.: **COM and .NET Interoperability**. Apress. 2002.

[Tufte 2003] Tufte, E.: **PowerPoint is evil**. In *Wired*. Issue 11.09, September 2003.
`http://www.wired.com/wired/archive/11.09/ppt2.html`

[Voswinckel 2003] Voswinckel, T.: **XML » SVG Presenter: Strukturierte Multimedia-Präsentation im Web**. Diplomarbeit FH Furtwangen. 2003.

[Watt & Lilley 2002] Watt, A. & Lilley, C.: **SVG Unleashed**. Sams. 2002.

[Watt 2001] Watt, A: **Designing SVG Web Graphics**. New Riders Publishing. 2001.

[Weidenmann 2002] Weidenmann, B.: **Erfolgreiche Kurse und Seminare**. 5. aktualisierte Auflage. Beltz. 2002.

[Weinreich, Obendorf & Lamersdorf 2004] Weinreich, H., Obendorf, H. & Lamersdorf, W: **HyperScout: Linkvorschau im World Wide Web**. In *i-com: Zeitschrift für interaktive und kooperative Medien* . Nummer 3, 4/2003. S. 4-12.

[Wexelblat 1998] Wexelblat, A: **History-Rich Tools for Social Navigation**. In *ACM Conference on Human-Computer Interaction (CHI'98)*. Los Angeles, CA. S,359-360.

[Wierzbicki 2003] Wierzbicki, R. J.: **The eLecture – In Search of A Better Way of Teaching**. In *Proceedings of EADIM Instructors Network Conference 03 Issues and Best Practices in Multimedia Teaching and eContent Training. 7.-9.* November 2003. Wien, Österreich.

[Wille 2004] Wille, C.: **Einführung in Performance Monitoring**. 2004. http://www.aspheute.com/artikel/20000428.htm

[Wolf & Steinmetz 2002] Wolf, P. & Steinmetz, A.: **How to capture a confererence - using the LectureLounge system to grasp the conference Experience**. In *Content Management - digitale Inhalte als Bausteine einer vernetzten Welt Tagungsband des Innovationsforum am 18. Juni 2002*. Fraunhofer Institut Arbeitswirtschaft und Organisation, Stuttgart. S. 81-90.

[Wolf, Putz, Stewart, Steinmetz, Hemmje & Neuhold 2004] Wolf, P., Putz, W., Stewart, A., Steinmetz, A., Hemmje, M. & Neuhold, E.: **LectureLounge – experience education beyond the borders of the classroom**. In *International Journal on Digital Libraries*. Volume 4, Number 1. 2004; S. 39-41.

[Zhang, Rui, Crawford, & He 2005] Zhang, C., Rui, Y., Crawford, J. & He, L.: **An Automated End-to-End Lecture Capturing and Broadcasting System**. Microsoft. Technical Report MSR-TR-2005-128.

[Zhu, McKittrick & Li 2004] Zhu, Z., McKittrick, C. & Li, W: **Virtualized Classroom – Automated Production, Media Integration and User-Customized Presentation.** In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*. 2004 S. 138- 138.

[Ziewer & Seidl 2002] Ziewer, P. & Seidl, H.: **Transparent TeleTeaching**. In *Online-Proceedings der ASCILITE 2002* http://www.unitec.ac.nz/ascilite/proceedings/papers/153.pdf

[Ziewer 2004] Ziewer, P.: **Navigational Indices and Full Text Search by Automated Analyses of Screen Recorded Data**. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004*. Washington, D.C., USA. S. 3055-3062.

[Ziewer 2006] Ziewer, P.: **Navigational Indices and Content Interlinkage on the Fly**. In *Proceedings of IEEE International Symposium on Multimedia 2006*. Workshop on Multimedia Technologies for E-Learning (MTEL), San Diego, CA, USA, 11.-13. Dezember 2006. S. 915-920.

[Ziewer 2007] Ziewer, P.: **Flexible and Automated Production of Full-Fledged Electronic Lectures**. Dissertation. Technische Universität München. 2007.

[Zimmerli 2004] Zimmerli, W.: **Brauchen wir eine europäische Lernkultur?**. Keynote-Vortrag auf der Learntec 2004. In *Elektronische Tagungsdokumentation LearnTec 2004:12 (CD)*. Europäischer Kongress und Fachmesse für Bildungs- und Informationstechnologie. Karlsruhe. 2004.

[Zupancic & Horz 2002] Zupancic, B. & Horz, H.: **Lecture recording and its use in a traditional university.** In D. Finkel (Hrsg.): *Proceedings of the 7th Annual Conferenceon Innovation and Technology in Computer Science Education*. New York: ACM Press. 2002. S. 24-28.

[Zupancic 2006] Zupancic, B.: **Vorlesungsaufzeichnungen und digitale Annotationen - Einsatz und Nutzen in der Lehre**. Dissertation. Albert-Ludwigs-Universität Freiburg. 2006.

# C. Abkürzungsverzeichnis

| | |
|---|---|
| ANT | Another Neat Tool |
| AOF | Authoring on the Fly |
| aofedit | Authoring on the Fly Editor |
| AOFwb | Authoring on the Fly Whiteboard |
| aof2Flash | Authoring on the Fly to Flash |
| ASV | Adobe SVG Viewer |
| BMP | Windows Bitmap |
| BMRC | Berkeley Multimedia Research Center |
| bzw. | beziehungsweise |
| ca. | circa |
| CD | Compact Disc |
| COM | Component Object Model |
| CSS | Cascading Style Sheets |
| DCF77 | Langwellensender zur Funkuhrensynchronisation:<br>D=Deutschland, C=Langwellensignal, F=Frankfurt, 77=Frequenz |
| DLH | Digital Lecture Halls |
| DOM | Document Object Model |
| DVD | Digital Versatile Disc<br>alternativ: Digital Video Disc |
| d.h. | das heißt |
| ECMA | European Computer Manufacturers Association (bis 1994, mittlerweile Eigenname) |
| EVES | Enhanced Viewer Experience System |
| evtl. | eventuell |
| FTP | File Transfer Protocol |
| ggf. | gegebenenfalls |
| GIF | Graphics Interchange Format |
| GNU | recursive Definition: GNU is not Unix |
| GPS | Global Positioning System |
| GZIP | GNU ZIP |
| HTML | Hypertext Markup Language |
| i.d.R | in der Regel |

| | |
|---|---|
| IIT | Indian Institute of Technology |
| jitl | Just In Time Learning |
| JPEG | Joint Photographic Experts Group<br>alternativ: Grafikformat |
| JSP | JavaServer Pages |
| LOD | Lectures on Demand |
| Lode | Lectures On DEmand: |
| MANIC | Multimedia Asynchronous Networked Individualized Courseware |
| MathML | Mathematical Markup Language |
| mlb | Multimedia Lecture Board |
| MP3 | MPEG-1 Audio Layer 3 |
| MPEG | Moving Picture Expert Group<br>alternativ: Bezeichnung für eine Familie von Multimedia-Datenformaten |
| MSI | Microsoft Installer |
| NTP | Network Time Protocol |
| o. ä. | oder ähnlich |
| PC | Personal Computer |
| PHP | rekursive Definition: PHP: Hypertext Preprocessor |
| ppt2aof | PowerPoint to Authoring on the Fly |
| ppt2mlb | PowerPoint to Multimedia Lecture Board |
| PRISM | Presentation, Replay and Interaction with Streaming Media |
| RSS | Really Simple Syndication (RSS 2.0) |
| SMIL | Synchronized Multimedia Integration Language |
| Stud.IP | Studienbegleitender Internetsupport von Präsenzlehre |
| SVG | Scalable Vector Graphics |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TTT | TeleTeachingTool |
| TU | Technische Universität |
| ULI | Universitärer Lehrverbund Informatik |
| VBA | Visual Basic for Applications |
| vgl. | vergleiche |
| Viror | Virtuelle Hochschule Oberrhein |
| VNC | Virtual Network Computing |

| | |
|---|---|
| VOD | Video on Demand |
| W3C | World Wide Web Consortium |
| WAV | Waveform Audio Format |
| WLan | wireless local area network |
| WWW | World Wide Web |
| XML | Extensible Markup Language |
| XSL | Extensible Stylesheet Language |
| XSLT | XSL Transformation |
| z. B. | zum Beispiel |

# D. Abbildungsverzeichnis

# E. Verzeichnis der Code-Beispiele

# F. Formelverzeichnis

# G.Index

## N



## O



## P



## R



## S



## T



## U



## V



## W



## X



## Z